DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(2)

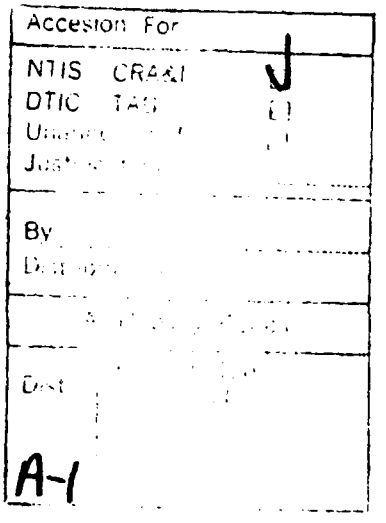
AFIT/GIR/LSY/88D-7

ANALYSIS OF THE HUMAN-SYSTEM INTERFACE USED IN THE PROGRAM MANAGERS SUPPORT SYSTEM: SOFTWARE COST ESTIMATING AND COMPETITION EVALUATION MODULES

THESIS

John A. Kane, Captain, USAF
AFIT/GIR/LSY/88D-7

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information is contained therein. Furthermore, the views expressed in the document are those of the author and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.



AFIT/GIR/LSY/88D-7

ANALYSIS OF THE HUMAN-SYSTEM INTERFACE USED IN THE PROGRAM MANAGERS SUPPORT SYSTEM: SOFTWARE COST ESTIMATING AND COMPETITION EVALUATION MODULES

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Information Resources Management

John A. Kane, B.S.

Captain, USAF

December 1988

Acknowledgements

This research effort could not have been completed without the generous contributions of many people who gave of their time and ideas. The original seed for this research was planted by Major Thomas Triscari. I thank him for his suggestion and his efforts as my reader.

I sincerely appreciate the sponsorship, support, advice, and encouragement that I received from Mr. Glen Miller and his staff from ASD/XRI. I also received valuable support, in the loan of a computer, from Mr. Ron Isemann and his staff from ASD/RM. Additionally, the time, efforts, and ideas given me by the many ASD people who served as test participants were invaluable.

My family played no small part in the completion of this thesis. My wife Kil Cha and my daughter Maria somehow found large amounts of patience and understanding to give me while I planted myself in front of my computer for weeks on end. Also, my parents gave me valuable support with their many prayers for my success.

I wish to thank Dr. Charles Fenno for his expert guidance during the initial formulation of this research. Finally, I wish to thank Lt Col Dorothy J. McBride for her expert guidance and many hours of wise counsel. Her strong hand at the wheel guided me through the uncharted, shoal waters of the unknown to the safe harbor of a completed thesis. To her I am indebted.

Table of Contents

List of Tables

Abstract

thesis

This ~~research~~ analyzed the human-system interface used in the Program Managers Support System (PMSS) software. The PMSS software is planned to consist of 18 modules that will run as an integrated system on standard Air Force microcomputers. Six modules are presently in prototype form. This research analyzed the human-system interface of two of those modules, the Software Cost Estimating and Competition Evaluation program.

Problem scenarios involving software cost estimation and competition evaluation were developed with the assistance of Aeronautical Systems Division (ASD) experts. Fourteen ASD program management persons were chosen as test participants (seven for each module) and attempted to work the problem scenario with the prototype software. The participants were only permitted to view the scenario and program documentation prior to the test. Their critical comments were recorded during the test session and later transcribed.

Transcribed comments were categorized and compared to expert guidelines published in the literature. Finally, suggestions for improvement in the human-system interface used in these modules were drawn from this comparison.

ANALYSIS OF THE HUMAN-SYSTEM INTERFACE USED IN THE PROGRAM MANAGERS SUPPORT SYSTEM : SOFTWARE COST ESTIMATING AND COMPETITION EVALUATION MODULES

I. Introduction

General Issue

The Aeronautical Systems Division (ASD) of Air Force Systems Command (AFSC) located at Wright-Patterson AFB, OH., is considering using the PMSS in its program management offices. ASD wishes to know how "useable" the PMSS is before it is implemented as a standard system within ASD. The term "useable" refers both to the functionality of the PMSS modules and to the quality of the human-system interface employed by the PMSS. This research assesses only the human-system interface quality and is intended to answer the question: "What aspects of the human-system interface can be improved to enhance user acceptance and utility of the PMSS within ASD program management offices?"

A human-system interface is not just a visual display on a computer system. The interface represents the totality of communications between the human user and the hardware/software system. It is an entity that provides a two-way dialog medium between the human and the computer.

When the interface is properly designed it can place the power of the computer into the hands of the human. The human can use this power to enhance and refine his/her cognitive efforts. When improperly designed the interface can confound and frustrate the human causing the user to direct valuable thought effort away from the problem at hand to system problem resolution. A good interface will be almost transparent and will help ensure adaptability to the largest number of PMSS users.

Specific Problem

As noted earlier, ASD needs to determine if the PMSS programs are useful to their program management activities. Since the domain knowledge of program management is possessed by ASD people and not this researcher they will determine domain satisfaction. This research will assess the quality of the human-system interface and combined with ASDs data will provide the answer to the usefulness of the PMSS to ASD.

The Program Manager's Support System (PMSS) is an application of decision support systems technology to the defense acquisition program

management environment. The PMSS is an evolving project conceived in-house in the Defense Systems Management College, Department of Research and Information, PMSS Directorate, Fort Belvoir, Virginia.

The purpose of the PMSS is to provide a management tool for managers in a program management office to assist them in their decision-making process and to help them execute their project in a more effective and efficient manner.

The PMSS is intended to support the defense Program Manager and his/her first echelon staff; for example, the Chief Engineer, the Plans and Programs Officer, the Configuration Manager, and the Integrated Logistics Support (ILS) Manager. The PMSS also can be used by other managers in the acquisition community, for example, by headquarters level executives, program management officers in major projects, and field activity managers.

The PMSS will be an integrated software system operable on various hardware systems. The target hardware is low-cost microcomputers although the system will also run on minicomputers. It will provide a capability to 1) integrate program management functional areas of responsibility, 2) generate program alternatives and impacts caused by various management actions and technical activities, 3) assess these impacts on the program management responsibilities, and 4) use other decision-making support methodologies. PMSS will provide educational tools to facilitate the teaching of program management functions at educational institutions involved with defense system acquisition program management.

The PMSS consists of two major parts, functional modules and the integrated PMSS. Functional modules are software programs that can be used as stand-alone programs to assist in program management areas of responsibility such as planning, acquisition strategy development, program management plan generations, cost estimating, scheduling, Program Objectives Memorandum (POM) development, budget generations, budget execution monitoring, financial management, systems engineering, production planning, integrated logistics support planning,

test issues identification, Test and Evaluation Master Plan (TEMP) generation, TEMP evaluation and monitoring, configuration management, document generation, document evaluation and monitoring, program office staffing and organization. The modules support specific functions of program management operations.

The integrated PMSS will provide a capability called Program Overview, which shows in a color-coded mode, the overall status of the program by the program hierarchical information categories. This provides the program manager an "instant" visual picture of his/her program status and quickly pinpoints program areas that require further management attention. The integrated PMSS looks across and within all functional areas of responsibility to assess the impact on the program and help the program manager develop alternatives for recovery.

The PMSS also will provide executive support aids such as briefing presentation aids, electronic mail, calendaring capability and telephone dialers. It also will include support capabilities such as word processing, spreadsheets, data-base management and decision tools (1:iii-iv).

DSMC states that the PMSS is not a management information system nor is it the decision-maker (1:iv). DSMC indicates the PMSS is a manager's tool to assist the program manager in his/her decision-making process. The PMSS will permit the integration of the user's experience, judgement and intuition to allow the user to evaluate available alternatives and, ultimately, aid the user to make better, more timely decisions. (1:iv).

Research Questions

The specific questions answered by this research are:

1. What procedures do PMSS module users employ to solve a representative problem scenario?

2. What human-system modifications should be made to each PMSS module evaluated?

Scope of Research

This research is necessarily limited to analysis of the human-system interface in those PMSS modules designated by ASD as having the highest priority. There are 18 PMSS modules (Appendix A) and an interface analysis of each of these modules would exceed the time and manpower constraints available to this research effort. Since the PMSS is an evolutionary system, only six modules are currently available and the integrated system will not be available in the near future. This research focuses on two modules: Software Cost Estimating (SWCE) and the Competition Evaluation Module (CEM). Concurrent research being conducted at ASD involves the Contract Appraisal System (CAPPS) and the Government Activity Tasking (GAT) module. The remaining two available modules have limited applicability at ASD. User evaluations are limited to ASD personnel who would be actual future users of the PMSS modules.

The approach used in this research is protocol collection in a field study to better test use of the modules in an actual working environment. Protocol analysis is based upon four theoretical assumptions:

1. The test subjects behavior can be perceived as searching through a defined problem space. The subject gathers knowledge, both correct and incorrect, about the problem as they proceed.

2. Each step that the subject makes requires the application of an operator,which is chosen from a finite set of task-related operators, to knowledge the subject holds in short term memory. The use of this operator injects new knowledge into short term memory and the subject is able to proceed forward in the problem space.

3. The vocalizations that the subject makes are related to some portion of the information currently held in short term memory. It usually directly relates to information that has just been acquired.

4. The information residing in short term memory, and vocalized by the subject, is primarily comprised of knowledge required as inputs to the operators, new knowledge produced by use of the operators, and symbols representing active goals and subgoals that are directing the activity (2:263-264).

The use of verbal data in the research process is not without its critics. The major criticisms are:

1. The reporting process might alter the performance of the task.

2. Reports may produce an incomplete record of the subjects cognitive processes.

3. Verbal reports are epiphenomenal. They are generated independently of the cognitive processes that produce non-verbal behavior and performance.

4. Verbal reports are idiosyncratic. They reflect unique experiences of individual test subjects and are not useful for general theory development.

5. Encoding of verbal protocols can't be made objective and sound because they are based on theory (2:169-170).

No claim is made that this research is not without its susceptibility to these criticisms although every effort was made by the researcher to minimize their effects. These efforts are outlined in Chapter 3.

Relevance of Research

This research has the potential of making a valuable contribution to the development of the PMSS as an integrated support tool for DoD program management offices. Persons actually engaged in program management activities at ASD solved representative problems with the prototype PMSS modules and provided detailed critical comments concerning the human-system interface. These comments, as categorized in this research, can be

used by DSMC to refine the PMSS into a tool that will be readily accepted and used in program management offices.

Introductory Overview

Chapter 2 of this research report details human-system interface issues as they exist today in the literature. There is a wide range of interface issues available, but only those relevant to the PMSS are included in this report. Specifically, hardware issues are not addressed because the PMSS is designed to run on the standard Air Force microcomputer.

Chapter 3 details the exact methodology followed in this research effort. It defines the warm-up task used, test site layout, description of equipment used, consent form used, a description of the problem-solving tasks, a description of test participants, and a description of the information collected during debriefing sessions.

Chapter 4 serves to categorize the critical comments collected during the field study and relates their importance to the PMSS. Chapter 5 contains recommendations based upon the information collected during the study. Transcripts of the dialog occurring during test sessions, problem scenarios used, and other relevant supporting documents are contained in Appendices.

II. Review of Literature

Introduction

This chapter analyzes the literature to form the knowledge base to help answer the research questions specified earlier. It examines the diversity that exists among computer users, a definition of an interface, the need for balancing automation and human control, interaction styles, dialog design, data display, use of color, screen layout, and issues related to program documentation. The first issue to be examined in the diversity among users.

Diversity Among Users

Basic to understanding the complexities of the human-system interface is to understand who the human user is. It is intuitively obvious that all humans are unique and possess a differing range of talents, intelligence, and physical abilities. Yet many software systems require the user to adapt to a rigid interface standard.

According to Shneiderman knowing the user is "a simple idea but a difficult goal and unfortunately, an often undervalued goal. No one would

argue against this principle, but many designers assume that they understand the users and their tasks" (3:53). Shneiderman classifies users into three groups: novice, knowledgeable intermittent, and frequent users (3:53-54). There is no specific standardization of user classification in the literature. Carey prefers to call them experienced, regular, and casual users (5:17). Though the nomenclature differs there is general agreement on relating degrees of semantic and syntactic knowledge to user levels.

Syntactic knowledge refers to the low-level device-dependent details required to operate a software system. Examples of syntactic knowledge include which key to press to delete a character, backspace, go to top or end of file, etc. This type of knowledge is complicated by two factors. First, syntactic functions vary from one system to another without any degree of standardization. Second, the user has a difficult time learning syntactic details because of their arbitrariness (3:47-48).

Semantic knowledge has two components: computer concepts and task concepts . Computer concepts include such things as how computers store, process, and retrieve information. Task concepts relate to the actual task at hand. In solving problems, users decompose the problem into a series of smaller tasks that when taken as an aggregate will solve the problem (3:49).

Integrating syntactic and semantic knowledge concepts is a formidable challenge to software designers. The syntactic/semantic user knowledge model can be used as an effective guide. First, semantics of the task objects should be specified and user actions clearly delineated. When this is done, the computer objects and actions are next identified. Only when both of

these steps are completed are the syntactic details dealt with. Following this formula software designs tend to be more comprehensible to the user and more device independent (3:52). To further comprehend these issues the human-system interface must be defined.

Definition Of An Interface

An interface is much more than a gateway to the software program. It is more than a simple screen or membrane separating the user from the computer. It is best thought of as a box that surrounds both the computer and the user and takes into account the user's intentions. The interface provides the user with tools to work with and the environment in which to work. Today's computer user desires an interface that is friendly or easy to use and that will support them in the task at hand. This implies an intelligent interface (4:377).

For an interface to be intelligent it must have access to different sources of knowledge. These sources of knowledge include:

(a) Knowledge of the User: this includes user experience, style, and preferences.

(b) Knowledge of the User's Task: This relates directly to the desired result of the user's efforts.

(c) Knowledge of the Tools (Available and being used): This refers to the protocols necessary to invoke a tool, their default parameters, side-effect, resources needed, and cost for usage.

(d) Knowledge of the Domain (User's task): This is highly specific to the application being used.

(e) Knowledge of Interaction Modalities: The various input/output displays, icons, color management, menus, windows, etc.

(f) Knowledge of How to Interact: When should the interface intervene? How long should it wait for user responses? When should it enter or leave tutoring modes?

(g) Evaluation Knowledge: An interface should know how it is doing in relation to the satisfaction of the user. It should be able to react to user dissatisfaction.

Including all of these sources of knowledge in an interface is exceptionally difficult but it offers a goal for programmers. An interface can be judged good or bad to the degree that it includes these sources of knowledge (4:378-381). The next issue to be examined is determining what the computer and human should do.

Balancing Automation And Human Control

The trend in interface design has been toward increasing automation. We see this expressed in the efforts of artificial intelligence in trying to imbue

the computer with human like thought qualities. Does this portend the end of the human user in the computing environment? Humans will always play a critical role because they function in the open system of the world. Computers operate in a closed system in that they can only handle a finite type and number of problems constrained by hardware/software limitations (3:75-76). To achieve some quality of balance between automation and the user we must understand those functions that each does best. Humans generally perform these functions better than machines:

(a) Sense low level stimuli
(b) Differentiate stimuli from a noisy background
(c) Recognize patterns in a changing environment
(d) Sense unusual events
(e) Remember principles and strategies
(f) Retrieve pertinent details without a priori connection
(g) Use past experiences to adapt decisions to different situations
(h) Choose alternatives if the first course of action fails
(i) Reason inductively
(j) React to emergencies and unusual situations
(k) Apply principles to solve varied problems
(l) Make subjective evaluations
(m) Focus on critical tasks when an overload occurs
(n) Adapt a physical response to a changing situation

Machines generally perform better at these functions:

(a) Sense stimuli outside the range of human abilities
(b) Count/measure physical objects or quantities
(c) Store quantities of coded information accurately
(d) Monitor prespecified events particularly those that occur infrequently

(e) Respond rapidly and consistently to inputs
(f) Retrieve quantities of detailed information accurately
(g) Process quantitative information in a prespecified way
(h) Perform repetitive actions reliably
(i) Exert highly controlled or great physical force
(j) Perform several activities simultaneously
(k) Maintain operations under a heavy information load
(l) Maintain constant performance over an extended period of time (3:76)

When brought to a problem situation the computer has almost all the knowledge it will need to perform (presuming the program matches the problem). It already knows how to execute the application but the human user often is not so well off. Software should be based on the concept of stepwise learnability, where the task is broken up into an easily comprehended series of steps. When this partitioning is employed the typical user will progress through five stages of learning:

(1) Using the system to perform a task with minimal required knowledge

(2) Learning the basics of the application system

(3) Progressing to an independent use of the system

(4) Investigating the more subtle or difficult system features

(5) Producing a quality product within known constraints (6:100-101).

An example of progressing through these stages of learning is the case of a new college student who has never used a text processing application program. The student knows he must learn to use this program to be

successful in school. The most pressing problem at hand, in the beginning, is to enter and print-out his notes from the first day of classes. Since he has no other assistance, he refers to the program documentation and learns how to open the application, enter sentences, and print them out. Since this level of knowledge allows him to satisfy the immediate need for notes he doesn't bother to learn any other program features. Later, the student must write a one page report and, by referring frequently to the documentation, learns how to do this. After several instances of having to write brief reports, the student can do this without needing the program documentation. Later, the student must write a thesis and learns how to use advanced features such as footnoting and headers/footers. He now knows how to use most or all program features and can write the thesis with minimal referral to documentation.

When a software system is designed to maximize the advantages of the user and the machine and to appreciate the user learning cycle, the probability of achieving a reasonable balance between automation and the user is increased. The next issue to be examined is the style of interface employed by the programmer.

Interaction Style

Five different primary interaction styles are available to be used by the programmer. In the literature they are commonly known as:

(1) Menu Selection

(2) Form Fill-in

(3) Command Language

(4) Natural Language

(5) Direct Manipulation

With menu selection the user is presented with a list of actions that are applicable to the task and chooses one of the available options. Using form fill-in a user sees a display of related fields and selects a field to enter data into. Command language allows a user to perform an action by entering a syntactical command into the system. Natural language is similar to command language in that a command is given to perform an action but differs in the syntax used. Natural language employs English language phrases common to human usage while command language requires a special computer syntax. Direct manipulation allows a user to perform tasks by manipulating visual representations of real world objects. This is the style used in the Macintosh computer and is increasingly found in the IBM environment.

Menu Selection. The greatest benefit from menu selection is its aid to clear decision making by the user. The user is presented with only a few choice alternatives and must choose from them. This style is most appropriate for the novice and intermittent user since it shortens the learning span and structures decision making. The user can be assisted with dialog that explains where the user is within the program and the applicability of the available choices to the next task at hand. Tasks are arranged in a top-down, hierarchical, fashion assisting in the selection of the next task. Menu selection does have disadvantages. Depending on the complexity of the task, menus may proliferate, in many layers, to the extent that the frequent user is frustrated. The frustration can be prevented by providing a method to bypass intervening menus. Menus also take up a great deal of screen space limiting the amount of information that may be displayed. Menus require a rapid display rate or else the user becomes frustrated waiting for the menu to appear (3:85-86, 8:577-578).

Much thought and research must go into effective menu design. Menus are typically arranged into one of three patterns: categorical (organized on the basis of meanings of menu items), alphabetic, and random. Existing research evaluating menu driven systems is based on search efficiency across these three patterns. This research has been inconclusive because the user's mental organization of the subject isn't taken into account (8:577). Hollands and Merikle cite a study by McDonald etal. (1983) suggesting that:

> A categorical menu organization was found to be superior to both alphabetic and random organization. An alphabetic organization might be

most appropriate when users do not have a well-formed mental organization of a particular subject domain. Conversely, categorical organization may be most effective when users have knowledge of the subject domain. In other words, a categorical menu structure may not be particularly helpful unless it is similar to a user's own internal mental organization (8:578).

Hollands and Merikle further advise that:

> First, the user group should be defined as precisely as possible in terms of knowledge expertise. Second, the particular tasks with which the users will be involved should be intensively examined. Depending on the outcome of these two considerations, appropriate organization strategies for particular sets of conditions are outlined in Table 1. For example, if novices to a subject domain are to be the users of a system and their main task is to be one of definition matching (searching for an assumed object or term), then categorical organization is recommended. If, on the other hand, experts are to be the users of a system and their task is to be one of term matching (searching for a given object or term), then either categorical or alphabetic organization would be appropriate.

Table 1: Recommended Menu Organizations For Different Tasks And Levels Of Expertise (8:585)

Task	Expertise	
	Novice	Expert
Term Matching	Alphabetic	Alphabetical or Categorical
Definition Matching	Categorical	Categorical

The selection mechanism used in menu systems is very important to the user. This concern can be further specified into numbered item and lettered item systems. When numbers are used items are clearly sequenced and the user can easily find numbers on a keyboard. This is particularly true if the user is not a competent typist. Numbering systems let the user see at a glance how many options are available. Sequential numbering aids in visual scanning and the user can make sure that each choice has been considered. The primary problem with numeric systems is that when there are ten or more items, two keystrokes must be made for a selection. Other problems include having different numbers for standard navigation commands, such as BACK TO MAIN MENU, on each screen. If natural numbering of menu items is not used the user might be misled to a preference for choice number 1.

If letters are used for menu items, then there is the choice between ABCDEF....lettering (sequential) and meaningful letter choices (mnemonic). Sequential lettering offers twenty-six choices before two keypresses are required. There is less chance of a keying error with letters than with numbers because the letters are more dispersed on the keyboard. It is more difficult for a nontypist to find letters versus numbers on the keyboard but this is not an insurmountable difficulty. Mnemonic lettering systems offer the advantage of linking a description of the task with the keypress. As examples, S can be used to SAVE and D for DELETE. If a task requires several menu selections, then provisions for a typeahead capability should be made. An experienced user could then press several menu selection sequences at once to accomplish the task rather than laboriously work his/her way through a series of menus. This capability allows one system to accommodate the novice and experienced user.

There are alternate strategies for making a choice with menu systems. These involve cursor movement schemes where the user moves the cursor around the screen by arrow keys, mouse, joysticks, tab key, or touch screen. These systems appeal to the novice user, even though more keystrokes are usually required, because of the satisfaction of being able to control the cursor. With these systems the selected item is highlighted both on the screen and the users mind. These systems do not allow for the use of typeahead schemes (3:117-119).

Shinar and Stern studied menu-driven systems and concluded that,

> A pointer-character override selection scheme should be provided as a means to address the needs of both novice and experienced users. Such

a scheme would allow the last character keyed to become the effective choice unless a pointer default selection is made with an entry key press (such as RETURN). In this scheme only one keystroke would be necessary for the character selection mode. Thus while the inexperienced user may move the cursor to the desired option and then press Return, the experienced user (familiar with the program and the options) would have only to press the appropriate letter or digit code (9:458).

Form Fill-In. Menu selection systems permit an easy selection of a choice but it is difficult to enter any data beyond a letter or number. A form Fill-in system allows users to enter data into many predefined fields. This is very similar to entering data onto the familiar printed paper form. In effect, the keyboard becomes a single menu from which many selections can be made. The user gains a sense of dialog control since all of the data is visible on the display. Shneiderman cites a study by Ogden and Boyle in 1982, in which a comparison of database update by form fill-in and a command language showed a significant speed advantage for the form fill-in style. Eleven of the twelve subjects stated their preference for the form fill-in approach (3:120-123).

There is a lack of empirical studies of the form fill-in approach but the following are suggestions for effective design:

- Meaningful Title: The topic should be identified without the use of computer terminology.

- Comprehensible Instructions: The user's task should be described in familiar terminology. Instructions should be brief and help screens should be used if more information is needed. The necessary actions should

be described briefly using terminology such as 'Type', and 'Press' (in reference to using special keys such as the TAB, ENTER, or cursor movement). The word 'ENTER' should not be used in instructions since if often refers to a special key.

- Logical Grouping and Sequencing of Fields: Related fields should be adjacent and aligned with blank space separating groups. Common patterns should be followed such as city, state, and zip code.

- Visually Appealing Layout of the Form: Fields should be uniformly distributed on the screen rather than crowding one section and leaving others blank. The user perceives proper alignment as representative of order and comprehensibility. If the user enters data from a paper copy, the screen should match the paper form. This allows the user to concentrate on the entry fields rather than the labels.

- Familiar Field Labels: Common terms should be used. An example of this is the use of the word 'Address' rather than domicile.

- Consistent Terminology and Abbreviations: The designer should prepare a list of acceptable abbreviations and terms and limit himself/herself to it. The abbreviation for any one term should not be varied with multiple use.

- Visible Space and Boundaries for Data Entry Fields: Underscores and other markers indicate the parameters of the space. The user can measure the intended entry and use abbreviations or other trimming strategies as needed.

- Convenient Cursor Movement: A simple method of moving the cursor should be used. The TAB key or cursor movement arrows are acceptable examples.

- Error Correction for Individual Characters and Entire Fields: The user should be able to make corrections or changes by use of the backspace key and overtyping.

- Error Messages for Unacceptable Values: The user should be shown an error message after the completion of the field if an unacceptable value is entered. The message should provide examples of permissible values.

- Optional Fields Should be Marked: Optional fields should follow required fields and be clearly marked as optional.

- Explanatory Messages for Fields: Explanatory information concerning a field should appear whenever the cursor is in that field. This information should appear in a standard location such as a window at the bottom.

- Completion Signal: When the user has completed filling in the fields it should be clear what to do next. Automatic completions should be avoided because the user may wish to go back and review or correct field entries (3:125-126).

According to Schneiderman,

> Columns of information require special treatment for data entry and for display. Alphabetic fields are customarily left justified on entry and on display. Numeric fields may be left justified on entry but then become right justified on display. When possible, avoid entry and display of leftmost zeroes in numeric fields. Numeric fields with decimal points should line up on the decimal points.

Other considerations in form fill-in design include dealing with multiscreen forms, mixing menus with forms, the role of graphics, relationship to paper forms, use of pointing devices, use of color, handling of special cases, and integration of a word processor to allow remarks (3:127-128).

Command Language. Command languages originated with operating system commands and are noted by their immediate reaction and impact on information or devices. The user enters a command and the computer system reacts. If the result is the one the user expects, he/she moves to the next step. If not, the user must adopt some other strategy. Command languages differ from menu selection systems in that command language users must remember notation and initiate the action. Menu selection users receive instructions and only need to recognize and choose from a limited set of visible alternatives (3:138).

According to Shneiderman,

> Command languages may consist of single commands or have complex syntax. The language may have only a few operations or thousands. Commands may have a hierarchical structure or permit concatenation to form variations. A typical form is a verb followed by a noun object with qualifiers or arguments for the verb or noun. Abbreviations may be permitted. Feedback may be generated for acceptable commands and error messages. Command language systems may offer the user brief prompts or they may be close to menu selection systems. Finally, natural language interaction can be considered as a complex form of command language (3:139).

According to Nickerson,

> Command languages differ greatly for different systems. Some of these differences are probably due in part to differences in system

capabilities. Some of them, however are not. One can find fairly large differences among command languages for systems that were intended to be used for the same purposes.

Languages have been designed, for the most part, in accordance with the intuitions of their designers. There is no theory of command language design or even a set of well-established design guidelines (10:477-478).

In an attempt to address this deficiency Shneiderman offers the following command language guidelines:

- Create explicit model of objects and actions.
- Choose meaningful, specific, distinctive names.
- Try for a hierarchical structure.
- Provide consistent structure (hierarchy, argument order, action-object)
- Support consistent abbreviation rules
- Offer frequent users the capability to create macros
- Consider command menus on high-speed displays
- Limit number of commands and ways of accomplishing a task (3:173).

Natural Language. Taking the command language style one step forward presents the possibilities of natural language interaction. With natural language interaction a user commands the computer using a familiar natural language (such as English). Complex command syntax doesn't have to be learned and choices don't have to be made from menus. The major problem with the natural language interaction style is its undesirability for large numbers of users for a wide variety of tasks. Computer implementation can also present problems but this is a much lesser concern. People differ from computers and what is right for human-human interaction is not necessarily right for human-computer interaction. Since

computers can display information many thousands of times faster than humans can enter commands, it is advantageous to use the computer to display large amounts of information and let the user choose among the items. (3:165).

Direct Manipulation. The last interaction style is direct manipulation. Using this approach the system designer creates a visual representation (such as an icon) of the action environment and the user directly manipulates the objects of interest. By pointing at these icons, users can carry out tasks and immediately observe the results. Direct manipulation is very appealing to novice users but can be very difficult to program. The next major issue to consider is dialog design.

Dialog Design

Whatever interaction style is used, it is essential that it resides on a foundation of good dialog design. The dialog design is the vehicle for communications between the user and the computer. Until fairly recently programmers placed dialog design in a position secondary to cost. Their main objective was to minimize equipment as well as personnel costs. This was dictated by the high cost and relatively limited speed of computing systems. This forced a tradeoff favoring the capabilities and limitations of the computer and not the human. Recent advancements in computer

technology and major reductions in equipment costs mean that designers should strive to optimize the human aspect of the interface (11:241).

According to Williges, et al,:

> In terms of human performance, the system designer must work toward an acceptably low error rate and an acceptable cost in personnel time. In addition, user acceptance and satisfaction with the computer system seem to be critical to effective use (11:241).

While there is no one set of universally accepted guidelines for dialog design, there are many generally accepted principles. The human factor guidelines for computer dialog designs are, not surprisingly, much the same as for any other type of application. Shneiderman has distilled these effective design principles into what he calls the Eight Golden Rules of Dialog Design. The literature generally agrees with these rules:

1. Strive for consistency: This is the most violated principle and the easiest to avoid. Systems should require the same sequences of actions for multiple similar situations. The same terminology should be used in prompts, menus, and help screens. Commands should be the same throughout the program. Any exceptions should be well thought out and constrained in number. Problems are easier to solve in a consistent environment. The user is able to develop a conceptual model of the system operation. Sometimes, in order to preserve consistency, systems may require operations that slow overall throughput such as requesting information that the system already has.

These situations should be limited and occur in a predictable manner (3:61-62,11:242,13:182, 14:257,15:39,16:116-122,17:3-14).

2. Enable frequent users to use shortcuts: As the user gains experience in system use, his/her desire to reduce the number of interactions and quicken the pace increases. Abbreviations, special keys, hidden commands, and macro capabilities are all of value to the knowledgeable user. Short response times and fast display rates also assist this category of user. Systems should be sufficiently flexible to adapt to or suit a variety of users with different skill levels (3:61-62, 11:242, 13:182,15:39,16:116-122,17:3-14).

3. Offer informative feedback: For every action that the user makes there should be some system feedback. The feedback involved will depend, in degree, on the importance of the action taken (3:61-62,11:242,13:182, 14:257,16:116-122,17:3-14).

4. Design dialog to yield closure: User action sequences should have a well defined beginning, middle, and end. The user is reassured when they receive informative feedback at the completion of a group of actions. The user can know that his/her intended actions were carried out and can move on to the next set of actions (3:61-62,14:257,15:39,17:3-14).

5. Offer simple error handling: To the maximum extent possible systems should be made "idiot proof" to prevent the user from making a serious error. Such an error might be the inadvertent erasure of a valuable document. If the user does make an error the system should detect it and offer advice for correcting it. The user should not have to retype the entire command sequence but should only have to retype the portion needing correction (3:61-62,13:182, 14:257,15:39,16:116-122,17:3-14).

6. Permit easy reversal of actions: The designer should acknowledge that users will make errors. Systems should allow users to reverse their actions that caused an error to occur. This will not be possible in all situations but this capability should be provided whenever possible. This ability to 'undo' an action relieves user anxiety and encourages them to explore unfamiliar system options (3:61-62,13:182, 14:257,15:39,16:116-122,17:3-14).

7. Support internal locus of control: Users should have the feeling that they are in charge of the system and that the system will respond to their commands. Systems should not surprise the user, require tedious sequences of data entries, and place obstacles in the path of a user trying to get information. Simply put, the user should be proactive and the system reactive and not vice versa (3:61-62,12:597,15:39,16:116-122,17:3-14).

8. Reduce short-term memory load: It is generally accepted that the limit of short-term human memory to be seven or eight items. Users should not be required to remember more than this number of commands, actions, or chunks of information. In situations where this limitation can't be adhered to, the system should offer assistance though on-line access to command syntax forms, abbreviations, or codes (3:61-62, 11:242,15:39,17:3-14).

Designers must also recognize the limits of the operators workload. It should be one of the first concerns when designing human-computer dialogues. Users will be prone to making more errors in overload situations so workloads should be kept within acceptable limits. Humans, in effect, act as single channel entities and when data from a variety of sources is

received it is queued until it can be processed The concept should be recognized when designing display screens to minimize scanning and information density (11:242).

Data Display

The standard display in most computers, particularly microcomputers, is the cathode ray tube (CRT) display. The user considers the display as essential to program use; it is the user's 'window' into the computer-based task operation. There are specific guidelines for various aspects of the display. Shneiderman cites Smith and Mosier offering the following five high-level objectives for displays:

1. Consistency of data display
2. Efficient information assimilation by the user
3. Minimal memory load on user
4. Compatibility of data display with data entry
5. Flexibility for user control of data display (3:69).

Displays can be separated into two broad categories: monochrome and color. Regardless of the technology employed the following variables are of concern:

- Size
- Refresh rate
- Capacity to show animation
- Resolution
- Surface flatness

- Surface glare from reflected light
- Contrast between characters and background
- Brightness
- Flicker
- Line sharpness
- Character formation
- Tolerance for vibration (3:258).

Each of the display technologies have advantages and disadvantages when each of these variables are measured. Displays should also incorporate the following features:

- User control of contrast and brightness
- Software highlighting of characters by brightness
- Underscoring
- Reverse video
- Character set (alphabetic, numeric, special and foreign characters)
- Multiple fonts (i.e. italic, bold)
- Multiple font sizes
- Shape, size, and blinking rate of the cursor
- User control of cursor variables
- Blinking at several rates
- Scrolling mechanism
- User control of number of lines or characters per line displayed
- Support of negative and positive polarity (light on dark or dark on light characters) (3:259).

Use Of Color

Color is one of the five visual input channels in a human being. Others are relative position, brightness, movement, and shape. There is no one set of simple guidelines for the use of color but the literature suggests the following precepts:

- Limit the number of colors used: No more than four colors should be used in any single display with a total of seven colors in a sequence of screens (3:338-342,18:51,20:119-120,21:44-45).

- Realize that color can be used as a coding technique: Color helps users accelerate task recognition. Color can be used to highlight those tasks of more immediate concern over routine tasks. It can also be used to speed the search for a specific item. From best to worst, the color speed search priority is red, blue, yellow, green, black, and white (3:338-342,18:51,19:549,20:119-120,21:44-45).

- Color coding should not confuse the task: Color should not be used capriciously. The decision to use color and the selection of a specific color should be closely linked to the task (3:338-342,18:51).

- Color coding should appear automatically: The user should not have to assign colors every time a program is used. It should appear automatically in direct support of the task (3:338-342).

- The user should be able to control color coding: The capacity to display color should be toggled by the user.

There are applications, usually text processing, where color can slow information comprehension or prove distracting (3:338-342).

- Program for monochrome first: Program designers should concentrate on logical screen design first and address color afterwards. Monochrome programs are more universally applicable due to the proliferation of monochrome monitors; the prevalence of color blindness in approximately 8 percent of the male population limits applicability of color displays; and the monochrome display serves to save the user's task if the color monitor fails (3:338-342,21:44-45).

- Color can help in screen formatting: When screen space is limited color can be used to group related data items and to distinguish similar but logically distinct data fields (3:338-342,21:44-45).

- Strive for consistency in color coding: The same color coding rules should be used throughout an entire system. Different uses of color within a system will only confuse users. This is particularly true when different programmers are used for separate modules within a system (3:338-342,21:44-45).

- Construct color coding rules around domain expectations: Program designers should confirm the rules for color use present in the task domain. For example, red means to stop for automobile drivers while it means a financial loss to accountants (3:338-342,18:51,20:119-120,21:44-45).

- Recognize the difficulty with certain color pairings: Some colors should not be used together on the same screen. As an example, pure red and

blue are on the opposite poles of the color spectrum and the human eye must strain to focus for both colors at the same time (3:338-342,18:51,21:44-45).

- Color changes should reflect status changes: When a data field is displayed in more than one color, the color change should serve to attract the user's attention to the change. For example, a routine hydraulic line pressure reading could be displayed in monochrome green and change to red when pressures exceed a safe limit (3:338-342,18:51,21:44-45).

- Use color in graphic displays to achieve greater information density: When multiple plots are made on one graph color can help the user distinguish the separate plots. Color is more effective in this than dotted lines, thick and thin lines, or dashed lines (3:338-342,19:549,21:44-45).

- Resolution is often poorer with color displays: Resolution is often greatly reduced when moving from a monochromatic display to a color display (3:338-342,20:119-120,21:44-45).

Screen Layout

A search of the literature does not provide a consistent set of rules for screen design and layout. A number of writers include an array of screen designs and identify one as better than another. The best and simplest guidelines for graphic design are described by Ives in an article citing Marcus. Marcus' six principles are:

1. Proximity: Objects that are located near each other will be perceived as belonging together.

2. Similarity: Objects that share a common attribute such as color, shape, or size will be interpreted as belonging together.

3. Common Fate: Object that were grouped together in one usage according to a specific pattern will be expected to conform to that pattern in follow-on uses. For example, a string of consecutive even numbers can be arranged in ascending or descending order without creating user confusion.

4. Objective Set: An object can assume a number of states but some of these states will be more strongly perceived by the user. For example, two lines that almost form a right angle will be perceived by the user as forming a right angle.

5. Direction: Objects that share a common direction will be perceived as belonging together by the user. Multiple lines in a line graph will have common entering and exiting directions. The user will use these directions to determine which lines interrelate.

6. Closure: The user will view objects as closed even when they are not. For example, a nearly complete circle will be perceived as a complete circle (21:27).

Documentation

Documentation serves as a road map for the computer user. It informs the user on how to use the system and to perform certain tasks. Accurate documentation will ensure that the system is used effectively and for a long time. Ineffective documentation will frustrate the user and cause him/her to not use a program or to use it ineffectively. Documentation initially serves the function of a training technique and with increasing experience later serves the user as a reference guide.

Documentation is categorized into six types. These forms are:

1. Analytical Documentation: Primarily used by the systems designer, it defines the needs and objectives of management that the program is to serve.

2. System and Feasibility Studies: These documents analyze the feasibility and cost factors of the proposed system.

3. System Documentation: This is mainly the systems specification that defines what the program is to do and how and when it will do it. Primarily of use to the designer and programmer.

4. Program Documentation: Produced by the programmer and specifying the logic and code used in the program.

5. Operations Documentation: Specifies all the procedures required for running the program.

6. User's Manual: This document explains to the user how to use the program. It provides conceptual and instructional information (22:98-99).

Of these six types the last two are of primary concern to users and the search of the literature is limited to them. Documentation can be written from several different view points. These viewpoints are:

- Analogical: The use of analogies to explain an idea. It explains a new idea based upon its relationship to a previously learned idea. This is a very powerful technique but it depends on the direct relationships between the two ideas. Educators commonly refer to it as moving from the known to the unknown.

- Analytical: Ideas are explained by the relationships of its facts. The approach is often linear and no comparisons are used. This approach is often used in the study of mathematics

- Empathetical: This approach presents ideas from the user's point of view and his/her 'sense of things'. It focuses on the user's decision making capacity and frame of mind. The writer of program documentation should consider the intended audience and choose an appropriate approach (22:99-100).

Shneiderman provides a list of guidelines for producing effective user's manuals. These guidelines fit neatly into each of the approaches discussed above:

- User's tasks guide organization (outside in)
- User's learning process shapes sequencing
- Present semantics before syntax
- Keep writing style clean and simple
- Show numerous examples
- Offer meaningful and complete sample sessions
- Draw transition diagrams
- Try advance organizers and summaries

- Provide table of contents, index, and glossary
- Include list of error messages
- Give credits to all project participants (3:373).

Documentation can be written in paper format or presented on-line. People are used to reading from paper and the literature suggests that their comprehension is better when paper, rather than on-screen, documentation is used (23:131, 24:342). On-line documentation offers several advantages over paper based systems:

- Information is always available when the user needs it. The user doesn't have to search for a misplaced manual or determine which is the correct book from a collection of manuals.

- The user's work space isn't cluttered with large or numerous paper manuals.

- Documentation can be updated electronically at greatly increased speeds and reduced costs over paper based systems.

- The user can quickly locate a needed section of information if electronic indexing or text searching capabilities are provided.

- The computer screen can display graphics and animation that are very useful in explaining complex ideas.

These advantages are offset by several potentially serious deficiencies:

- Computer displays are not as readable as are printed manuals.

- Displays show less material than paper manuals. Resolution is poor in comparison and paging is much slower than turning a page.

- Most people are very familiar with the organization and use of

printed materials. They must learn a new set of skills to navigate through an on-line system.

- RAM memory can be overburdened if the user must toggle back and forth between his/her work and on-line documentation. The user also loses his context of the work and this can contribute to forgetting instructions from the on-line documentation. This can be mediated by the use of multiple screens or windows (3:374-376).

Summary

It is clear from the review of literature that designing an effective user interface is a very complex task. The designer must first understand the diversity that exists among users, what an interface is, the need to balance automation and human control, the various interaction styles, the principles of dialog design, and the issues concerning data display. The designer must also understand the principles underlying the use of color, how to construct effective screen layouts, and how to write documentation that truly helps the user.

III. Research Methodology

The method of research used for this project is observation (protocol collection) and debriefing of PMSS system users involved in solving problems (Appendix B and C). Test subjects were selected by ASD work centers with the criteria that they be interested, as potential users, in the PMSS modules being tested and sufficiently computer literate to be able to use a keyboard. Seven test participants were selected for each PMSS module tested. A greater number of participants were desired but this number was selected because of limited participant availability, availability of only one computer and one room cubicle, and researchers time constraints imposed by AFIT.

Subjects were provided with program module user's manuals and problem scenarios during a series of pre-study briefings. Both scenarios were developed with guidance from an ASD program management expert experienced in software cost estimating and evaluating the effects of competition on an acquisition. The ASD expert reviewed and certified each scenario as representative of a typical problem that users might encounter in their working environment. The original intent was to brief all participants during one mass briefing to minimize any variations in instructions. This was not possible because many test subjects had work schedule conflicts with the scheduled briefing. Other subjects were on extended temporary duty to other locations and were briefed as they became

available. The original briefing was recorded on tape and this was used to construct a briefing agenda (Appendix D). This agenda was used during a second mass briefing and at all subsequent individual briefings to ensure consistency in instructions.

As part of the briefing, subjects were provided with a brief description of the PMSS modules being tested in this and concurrent research and asked to rank order their module preferences from 1 to 4, with 1 being the most preferred and 4 the least preferred (Appendix E). This was done in an attempt to match the subjects desires with the module actually tested. Every effort was made to test individual subjects on their most preferred module although this was not possible in some cases because of the need to equally distribute test participants among four software modules (two modules in this research and two in concurrent research) and the interferences of test participant temporary duty assignments. This researcher could discern no negative effects from this situation since all test participants appeared highly motivated and the direction of this research was the human-system interface, not domain knowledge.

In no case was a subject tested on a module that they had prior knowledge of. In all cases at least one week separated the initial briefing and test period to permit subjects to read the problem statements and program documentation.

Subjects were asked to read the information presented in these documents at least once before their test period. They were not permitted to use the software prior to the protocol collection. A question to this effect was included in debriefing sessions to ensure that this restriction

was observed. During the problem solving activity their comments were recorded using a SANYO microcassette recorder (Model 5495). After the test session they were debriefed to capture any additional reactions.

A checklist was used during test sessions to ensure consistency of test procedures (Appendix F). This checklist also includes those questions posed during debriefings. Every effort was made to ensure that test subjects were not interrupted by noise, telephones, or other similar influences. This was not totally possible because of the working office environment that the test area was located in. A diagram of the test area is included as Appendix G. Office space is very limited at ASD and the researcher was fortunate to obtain an area for testing. The only environmental noise that may have affected test subjects were general office noises (telephones, muffled talking, etc.) that escaped office cubicles and entered the test area over the six foot wall panels. In no case did the researcher note any outwardly visible effects of this noise on subjects during test sessions.

All testing sessions were conducted in Bldg. 11A, Room 201 using a Zenith Z-248 microcomputer configured with one 360K floppy disk drive, a 20 MB hard disk drive, enhanced graphic adapter card, and color monitor. PMSS software was loaded onto the hard disk by the researcher and booted to the opening screen for each participant. The floppy disk drive was not used. This was a new machine obtained on loan from ASD/RM for the purposes of this test. The researcher made every attempt not to talk during test sessions and was present in the room. It was discovered early in the first test session that the researcher had to ask questions of the subject in

order to elicit responses. Test subjects tended to refrain from vocalizing their thoughts unless an occasional question was asked of them. This problem occurred even though a warm-up task was employed to give the participant practice at vocalizing thoughts prior to the actual test. The warm-up task consisted of playing the public domain game 'NYET' for approximately five minutes. This task was chosen because the game employs color pairings, windows, system messages, and screen designs to help the test participant begin thinking about human-system interface issues. Complete transcripts of all testing sessions are included in Appendix H. Test subjects were guaranteed anonymity in writing (Appendix I) to promote open and honest reactions. Test participant comments were transcribed, categorized, and compared against human-computer interface guidelines reported in the literature review.

Specific steps taken were:

1. Briefed test participants about the purpose of the test and specific procedures. Scheduled specific testing periods. Distributed user's manuals and problem scenarios.

2. Test participant spent 5-10 minutes at the warm-up task. The public domain game program 'NYET' was employed as a warm-up device. During and after this play, the researcher posed general user interface questions to the participant. This was intended to help the participant practice vocalizing their thoughts and to place them in a frame of mind to think about interface issues. These questions are included in Appendix E.

3. Administered problem scenario and record comments for all test participants. Debriefed test participants.

4. Transcribed all dialogue recorded during test sessions. Categorized them by the major interface concerns reported in the literature review for each step in the problem.

5. Made conclusions for each interface concern encountered during testing. More than two test participants commenting, either positively or negatively, on the same interface issue was considered significant. Comments not meeting this criteria are included in Chapter 4 under a separate heading.

Chapter 4 presents the categorizing of the collected data. Chapter 5 presents the researchers conclusions inferred from this data with recommendations for further research.

IV. REPORT OF RESULTS

This chapter presents an analysis of the data gathered from the test participants. Because the two PMSS modules tested are dissimilar in nature and designed to operate in a stand-alone configuration, this analysis is divided into two major sections.

The first section deals with the Software Cost Estimating module and the second section addresses the Competition Evaluation module. Each section begins with a description of the computer experience level of the test participants to help comprehend the complexity, or lack thereof, of their critical comments.

Each section is further divided into task numbers that directly correspond to the task numbers in the problem scenarios. This system helps to summarize the comments made by the test subjects and to note similarities. Specific comments given by the participants are not restated here since transcripts are included in Appendix H.

Software Cost Estimating Module

Seven test participants worked the SWCE module. All of the participants are involved in ASD program management activities. Five of the seven are

directly involved, on a daily basis, in software development cost estimation. Of the two remaining participants, one (Respondent 2) is a program manager and supervises people who do software cost estimating. The other participant (Respondent 3) was familiar with the subject but is not actively involved in software cost estimating processes.

Three of the participants (Respondents 3, 4, and 5) can be considered as expert computer users since all possess graduate degrees with heavy orientations in the computer sciences. One person (Respondent 3) is in the process of completing doctoral requirements in computer engineering. The other four participants can be considered as moderately experienced computer users since they use computers extensively on-the-job. All of these people have had training in applications software or computer programming.

All of the test participants stated that they had never seen the SWCE module prior to the test session. One person (Respondent 3) indicated that he had previously received a briefing about the SWCE module but felt he had received no information that would have prejudiced his comments. Three persons (Respondents 5,6, and 7) had seen other PMSS modules prior to their SWCE test session. Since there is no standardized interface used in all PMSS modules, no prior learning curve can be assumed for these persons. Respondents 1,2, and 4 had no prior exposure to any PMSS modules.

All of the participants stated they had studied the problem scenario and documentation prior to their test session. Study times ranged from a high of two hours to a low of 15 minutes. On the average, participants spent approximately 55 minutes studying the scenario and documentation.

Given this background, the critical comments of the test participants can be considered as credible and in some instances very well informed.

Task One. The first task required the test participants to open an existing SWCE database, select an operation to perform, and change several existing values in the database.

Five of the participants (Respondents 1,2,3,4, and 6) expressed dissatisfaction with the procedures required for loading a data file and adjusting parameters. From the main menu the participants selected 'Alt-L' to access the list of data files, moved the cursor next to the desired file, placed a number,beginning with one, next to that file name, entered 'return', returned to the main menu, chose 'Alt-A' to adjust parameters, chose 'Alt-C' to change ratings, selected a rating from a sub-menu, used F6 to change windows, and entered numeric values. Participants felt that the use of Alt key combinations required too many keystrokes. They stated a preference for either a single key command or the use of function keys. Participants did not like having to enter a number next to the name of the data file they wished to load. They indicated a preference for moving the cursor over the name of the data file and entering a 'carriage return' as this is the technique most commonly used in other popular software programs.

Three of the participants (Respondents 3,5, and 6) expressed displeasure with the on-screen instructions (Appendix J). Respondents 3 and 6 felt that the instructions should have a toggle for turning them off and on. Their rationale was that the instructions were useful for first time users but

would become distracting to the frequent user. Respondent 5 felt that the instructions were too long and preferred a series of short bullet statements versus full paragraphs.

Four of the participants (Respondents 2,5,6, and 7) liked the system prompt that advises the user to "Please be patient" while values were recalculated. They felt reassured that the computer was actually working and not malfunctioning.

Three participants (Respondents 3,4, and 6) complained about the way color was used in the SWCE program. All of them felt that the existing white and light blue lettering were too close to differentiate values that were changeable from those that were not. They expressed a desire to be able to select colors more to their own liking from a color palette.

Task Two. Task two simply required the participants to save the revised data in a new data file and give it a file name. Only two areas of user dissatisfaction were noted. The first involved the use of 'Alt-E' for saving the data file. Respondent 5 felt that 'Alt-S' should be used for the save function no matter where in the program the file action occurs. Alt-S is presently used to select the sensitivity analysis function. Respondents 4,6, and 7 felt that the system response time to save such a small data file was inordinately long. Once again, the 'Please be patient' system message that was displayed during the save function was noted positively (Respondent 1).

Task Three. Task three required the test participants to access the sensitivity analysis function and to produce a graph of the 'CPU Time Constraint' driver with respect to development time. The graph was then to be adjusted to full screen size.

The need to have 'Symphony' software in order to run the SWCE module was noted negatively by Respondents 5 and 6. Few of the test subjects have access to Symphony in their offices. Most use Lotus 1,2,3 or Enable. At any rate, the participants felt that SWCE should be written as stand alone software.

Respondents 3 and 5 felt that the on-screen instructions for the sensitivity analysis function were verbose and tended to confuse them. Instructions were presented on the first screen that pertained to following screens and the participants felt they had difficulty remembering them.

When the user selects the cost driver name for sensitivity analysis only the first word of the name is highlighted by the cursor. Respondent 5 felt that this was confusing and suggested a multi column list. The entire cost driver name should be highlighted to avoid this difficulty.

Screen flashing (as if the screen is turned on and off) as the graph is drawn was noted as a major difficulty by Respondents 2,3,6, and 7. The screen flashed a total of three times as the graph was drawn and this startled the users. The general feeling among these respondents was that the screen should go blank as the graph is being calculated and should only display the graph when it is ready.

Several positive aspects of the graphic sensitivity analysis display were noted by Respondent 3. These included good axis labeling, clear display of each data point, and the fact that the graph did not originate at zero so no resolution was lost (Appendix K). A suggestion was made that guide lines extend from the data points to the axis so exact value determinations can be made.

A major difficulty noted by Respondents 2,3,4, and 6 was the lack of instructions on the full screen graphic display. They did not know how to leave the full screen display and return to the sensitivity analysis menu, and the use of the 'return' key was not intuitively obvious. All expressed a desire for on-screen instructions to alleviate this problem.

Task Four. Task four required the participants to access the 'What-if Analysis' function, select two data files, produce an activity distribution graph of these files, and adjust the graph to full screen size.

Respondent 3 had difficulty in leaving the sensitivity analysis function and going to the 'What-If Analysis' menu. On screen menus did not clearly present the available choices as they blended into the screen because the letters are white and the background is light blue.

Once again, the requirement for Symphony was noted as a deficiency by Respondents 3 and 5. They suggested that SWCE be written in a commonly used language as a stand-alone module. This would allow portability to a large variety of machines and eliminate the need to buy Symphony.

The capability to graph more than one data file at a time was noted as a positive aspect of the SWCE module (Respondent 2). This provides a powerful comparison tool for management.

Sluggish processing was noted once again (Respondent 3), and the use of Alt key combinations was considered tedious by Respondent 5. Adding the capability for using a mouse was requested by Respondent 5.

The user can enter a name for a data file in all caps, lower case, or mixed caps and lower case. When the data file is graphed the file name is displayed on the screen in the same fashion. Respondent 3 felt that the name should be displayed in all caps and the program should adjust any user entered name into all caps. The display of the data file name on the graph was noted positively by Respondent 1 who felt that the name of the current data file should be displayed on all screens throughout the program.

Screen flashing as the graph was being calculated was found distracting by Respondents 2 and 3.

A desire to select colors for graphic displays was expressed by Respondents 3 and 5. The currently available colors of green, yellow, and red suggest varying degrees of risk in the program management area. A desire for enhanced graphics capabilities such as shadowing, and 3-D effects was expressed to help enliven graphics displays (Respondent 3). Axis scaling should also be in commonly used units such as days, weeks, months, and years rather than thousandths of a manmonth (Respondent 3).

Several test participants (Respondents 3,4,5, and 6) noted that SWCE graphic displays are in color yet few people have color printers. Their

prognosis was that the availability of color printers was very limited and would remain so in the future. They expressed a desire for cross hatching capabilities so that graphic displays could be printed on readily available single color printers.

Once again, participants had difficulty in leaving full screen graphic displays (Respondents 2 and 5). Respondent 3 suggested that the word 'Zoom' not be used in the menu to describe a full screen display. Use of the words 'Full Screen Display' was considered a better choice.

SWCE Summary

In summary, test participants felt that the color graphics capabilities were the best feature of the SWCE module. Major drawbacks were the required use of Symphony, inconsistent command key combinations throughout the program, screen flashing while graphs are calculated, and the lack of on-screen instructions for leaving full screen graphic displays.

All respondents felt that the SWCE program documentation manual was well prepared and a definite aid to effective program use.

Competition Evaluation Module

Seven test participants worked the CEM module. All of the participants are involved in ASD program management activities. All of the participants could be classified as moderate users since they use computers daily on-the-job and have no extensive computer training.

Six of the participants stated they had never seen any PMSS module prior to the test session. Respondent 13 indicated he thought he had seen the CAPPS module before but was not certain. There did not appear to be any prior learning that would have affected test results.

All of the test participants studied the problem scenario and documentation prior to the test session. Time spent studying these documents ranged from a high of two hours to a low of two minutes. On average, participants spent 55 minutes with the scenario and documentation.

Given this background, the critical comments collected from the test participants can be considered informed and relevant.

Task One. Task one required the test participants to create a new data file and to enter three data values. Three of the participants (Respondents 10,11, and 12) felt the opening instruction screens were too long and too compact. Respondent 10 felt that this could be overcome by displaying the section titles in color. At present, the CEM module has no color capabilities. Respondent 11 suggested that the opening screens only include the purpose of the module and not the underlying theory.

Task Two. Task two required the test participants to input eight data values in response to program prompts. Six of the participants (Respondents 8,9,10,11,12, and 14) were confused by the 'First Unit Cost' being in millions of dollars. The value that they had to input was less than one million dollars which required it to be entered as a decimal figure. The participants felt that the program prompt did not clearly communicate that the cost was in millions. Two of the participants (Respondents 8 and 10) felt the program should allow cost figures to be entered directly regardless of their values. Respondent 11 was the first to note that the program did not allow reverse scrolling to correct errors.

Task Three. Task three required the participants to enter eight data values in response to program prompts. The major problem encountered in this task was confusion caused by unclear program prompts. Respondents 8, 10, 11, 12, and 14 all expressed dissatisfaction in this area. Because no color is used and menus are in a compact horizontal format, menu choices tended to blend into the entire screen and confused the participants. Respondent 3 noted a program inconsistency when percentage values were asked for. In task one percentage values were displayed with a percent sign that was automatically entered by the program. In task two percentage values were not displayed with a percent sign. This is not a major problem but was found disconcerting by the test participant.

Task Four. Task four required the test participants to enter identical values for a second competitive source that had been entered for the first competitive source in task three. Two possible courses of action were

available: values could be entered figure by figure or the 'return' key could be used to automatically enter the value. Respondent 8 chose to enter the individual figures rather than use the 'return' key as a means to learn the program. Respondent 9 did not understand the on-screen instruction using the symbol 'CR' for carriage return. Respondent 11 did not understand the on-screen instructions for using the 'return' key and Respondent 14 did not notice the availability of the automatic entering feature.

Once again, this indicates that program prompts and instructions are not written and arranged in an easy to comprehend manner. Respondents 10,12, and 13 made use of the 'return' key and thought this was a very desirable program feature.

Task Five. Task five required the participants to enter seven data values establishing the timing of shift and rotation due to competition. Six of the participants had no difficulty with the task and offered no comments. Respondent 8 experienced some confusion with the program prompt 'Downward Shift(% or carriage return if none)'. This person stated a preference for 'Downward Shift (% or if none, carriage return)'.

Task Six. Task six required the participants to enter six data values concerning cost minimization, non-recurring costs, other recurring costs, and inflation rates. All seven respondents had no difficulty with this task and offered no comments.

Task Seven. Task seven required the participants to name and save the data file, display the data, and end the analysis.

All seven respondents had difficulty in choosing the proper command for saving the data file. The menu choice is presented as '(F)ile Out' and none of the participants understood what that meant. To a person, all expressed the preference for 'Save' since a wide variety of programs uses this command for saving a data file.

All of the respondents liked the way the data display was presented on-screen. They viewed it as a very desirable program feature because it was comprehensive (Appendix L).

Task Eight. Task eight required the participants to call-up the data file they had previously created and to make fourteen data value changes.

Several problems were noted by the respondents during this task. The first occurred because no index of data files is presented on-screen. The participants had to call-up their data file name from memory. Respondents 8,10,12,13 and 14 all expressed a desire for a data file index.

The second problem occurred because of inconsistencies in the user interface. In previous sections of the program commands were the first letter of the command word, i.e. '(F)ile Out'. In this section of the program menu choices are presented as A,B,C, and D. This momentarily confused Respondents 10 and 11.

The next problem involved the long, compact, horizontal format menu. All respondents reported difficulty in reading the menu and stated a preference for a vertically formatted menu with spaces between the choices. Respondent 13 expressed a desire for color to help differentiate menu choices.

Another problem occurred when the participants had to change a data value for only one of the five years involved in the scenario. The menu choice is presented as '(Y)ear (1)' meaning only one year. Respondents 8 and 10 had difficulty with the menu choice because they interpreted it to mean the first year.

Respondents 9,10,12 and 14 experienced difficulty with the prompt, 'choose shift or rotation between one and one'. This message occurred because only one shift and rotation had been entered earlier. These respondents were confused by the message and preferred the program to take them directly to the year when only one had been entered.

Task Nine. Task nine required the participants to name and save the changed data file and to quit the program. Respondents 10 and 12 noted interface inconsistencies because 'Q' was used for 'quit' where 'E' for end had been used previously in the program.

CEM Summary

In summary, the major problems noted in the CEM module were long and hard to read opening instructions, a lack of color capability, confusing menu prompts, no capability to scroll backwards to correct errors, and inconsistencies in commands throughout the program. The most appreciated program features were the ability to automatically reenter values that were not being changed and the arrangement of the data summary.

This chapter has reported the procedures used by the test participants in solving a representative problem scenario and as such answers the first research question: What procedures do PMSS module users employ to solve a representative problem scenario?

The next chapter will compare the major problems noted by the participants with the guidance given by the experts in the literature review and will suggest improvements.

V. CONCLUSIONS AND RECOMMENDATIONS

Conclusions

This research was intended to serve as an assessment of two PMSS software modules to determine if they could be useful to ASD program management activities. ASD presently uses other programs or individually developed spread sheet applications to solve software cost estimating and competition evaluation problems. The question then, from the ASD point of view, is if it is better to use the SWCE and CEM modules or stay with their present methods.

Though the sample size of test participants was small it was representative of ASD's needs because it involved the actual people who work these types of problems on a daily basis. These people know intimately the problems associated with their present methods of doing business. This fact, I believe, was directly responsible for the high level of interest and obvious enthusiasm for testing the PMSS prototype modules.

Based on the results of these tests I must conclude that the SWCE and CEM modules are useful to ASD program management activities. The SWCE program provides color graphics capabilities that are presently unavailable in other software packages in use. The CEM program offers effective data summaries and a menu based interface that can be used even by inexperienced program management people.

Both the SWCE and CEM programs are prototype models and fulfill their intended function particularly in the semantic domain. They do return an appropriate answer when the user enters the proper data. As Shneiderman has noted it is a very difficult task for software designers to integrate syntactic and semantic concepts. It is in the syntactic domain that improvements can be made to both modules that would greatly increase their ease of use by program management workers.

Every comment noted in Chapter IV is not individually addressed in this chapter as it would be tedious to do so. Comments given by the test participants concerning dialog design issues were combined and compared to Shneiderman's Eight Golden Rules of Dialog Design. This taxonomy was chosen because it distills guidance not only from Shneiderman but several other experts as well. These experts are also identified in the recommendations for each module.

Recommendations (SWCE Module)

A significant improvement could be made to the SWCE program by eliminating 'Alt' key combinations for program commands. These combinations violate the second of Shneiderman's Eight Golden Rules of Dialog Design and the guidelines established by Williges and Williges, Ting-peng, Lantz, and Simpson by failing to provide shortcuts. It requires two keystrokes to execute a command with 'Alt' key combinations. As noted by

Shneiderman, two alternate strategies are available. These are to use numbers or letters for command choices. Numbers would limit the number of choices before two keystrokes were required. Letters would offer twenty-six choices; a far greater number than used by the SWCE module. Letters also reduce the chance of a keying error due to their dispersion on the keyboard.

Using a lettering scheme for menu choices would also help solve the the inconsistency of commands problem which violates the first golden rule and the guidance of Williges and Williges, Ting-peng, Norman, Lantz, and Simpson. The letter 'S' could be used for saving a data file no matter where in the program the function was performed. This would eliminate the use of 'Alt-S' in one part of the program and 'Alt-E' in another.

Shneiderman's seventh golden rule and the advice from Rushinek and Rushinek, Lantz, and Simpson, to support the user's internal locus of control is violated when the screen flashes during graph construction. As noted in Chapter 4, this flashing startled users. Screen blanking or a 'Please be patient' system message would be preferable to the screen flashing.

The user's internal locus of control could be strengthened by providing a capability to turn on-screen help files on and off. As noted in Chapter 4, test participants recognized that not having this capability would tend to place an obstacle in their path as their familiarity with the program increased.

As noted by Shneiderman, Durrett and Trezona, Tullis, Davis and Swezey, and Ives, the user should be able to control color coding. Also, there are

difficulties with color pairings that are close in value to each other. Both of these deficiencies were noted by test participants in the SWCE module. A capability for user selection of colors should be incorporated in the program.

These recommendations for improvement to the SWCE program are directly supported by the test data described in Chapter 4 and the expert's guidance for these matters included in Chapter 2. It is reasonable to conclude, if these improvements were made, the value of the SWCE module to the user would be significantly enhanced.

Recommendations (CEM Module)

The most major improvement that could be made to the CEM module is to provide a capability to permit easy reversal of actions. This capability is supported by Shneiderman's sixth golden rule and the guidelines of Ting-peng, Norman, Lantz, and Simpson, and recognizes the fact that users will make errors. Currently, the only way a CEM user can correct an error is to proceed to the data summary display and employ an error correction sub-menu. The capability should be added to reverse scroll anywhere within the program and to type over any errors. A second alternative for error control would allow the cursor to be moved to any location within the data summary display and any data value changed in much the same manner as used in spread sheet programs.

The second most important improvement that could be made to the CEM program is to clarify the confusing menu prompts and to provide consistent program commands. Both of these items are discussed in Shneiderman's first rule and the guidance of Williges and Williges, Ting-peng, Norman, Lantz, and Simpson. An item such as '(Y)ear (1)' could easily be replaced with 'Choose one year' and would eliminate user confusion. It is evident from test participant reactions that 'Save' should be used for saving a file instead of '(F)ile out'. A number of instances such as these are documented in Chapter 4 and could be easily changed.

It was noted in Chapter 4 that the opening instruction screens were found long and verbose by test participants. Shneiderman has addressed this area in his eighth golden rule, as has Williges and Williges, and Lantz, of reducing short-term memory load. The underlying theory employed in the CEM module might be more effectively placed in the documentation rather than in the program. A simple use of more white space would also help the user read the instructions without feeling that they are too long.

While it may be outside the original requirement of the CEM program, the addition of color would be a significant enhancement. As noted by Shneiderman, Durrett and Trezona, Tullis, Davis and Swezey, and Ives, color can be used as a coding technique to help the user highlight tasks of immediate concern. The use of color would help the user identify task particularly in the latter stages of the program when menus tend to get long and run together. The addition of white space in these menus would also help reduce user confusion.

These recommendations for improvement to the CEM program are directly supported by the test data described in Chapter 4 and the expert's guidance for these matters included in Chapter 2. It is reasonable to conclude, that if these improvements were made, the value of the CEM module to the user would be significantly enhanced.

Summary

As noted earlier, both the SWCE and CEM modules are useable as they presently exist. Since both of these programs are in the prototype stage it is appropriate to offer suggestions for improvement. As a point of fact, DSMC has solicited recommendations for improvements. The suggested improvements offered above are based in solid, expert guidance as discussed in my literature review. They are improvements the users desire and this has been documented in Chapter 4. This combination of expert guidance and documented user desires serve to answer the second research question: What human-system modifications should be made to each PMSS module evaluated?

Further Research

As noted in Chapter 1 the scope of this research was limited to just two PMSS modules. Concurrent research is addressing two more modules. Both of these efforts combined only investigate four of a planned total of 18 program modules. As more PMSS modules become available in prototype form further research should be performed on the human-system interface. When all 18 modules are available, further research should be conducted to determine the accuracy, consistency, and efficiency of all 18 modules in an integrated form.

Appendix A: List of Planned PMSS Modules

Module Name	Brief Purpose
Procurement Strategy Module	Select a procurement strategy
Schedule and Resource Allocation	Develop Gantt Chart schedule Do resource allocation
Contract Appraisal System	Monitor Contracts
Schedule Risk Assessment Management	Network development Milestone management Schedule risk assessment
Software Cost Estimating	Develop cost estimates for software development
Program Office Organization and Staffing	Develop PMO organization charts Administer staffing functions
Government Activity Tasking	Generate and monitor tasking of government activities
Documentation Configuration Control	Monitor documentation status

Automated Program Planning and Documentation	Generation and monitoring of ASP/PMP, PS/P, TEMP, SEMP, ILSP, RAMP
Parametric Cost Estimating	Conduct parametric cost estimates
TEMP Auditor	Evaluate TEMPs
Project Control System	Conduct PMO financial management
Budget Preparation and Execution	Develop budgets Monitor progress
Test Issues Management Evaluation	Conduct pre-TEMP planning
Cost Analysis Strategy Assessment	Conduct LCC analysis
Venture Evaluation Review Techniques	Conduct risk analysis
Quick Cost	Conduct quantity/cost trade-off analysis
Competition Evaluation Module	Conduct production competition analysis

Appendix B: Software Cost Estimating (SWCE) Scenario

You are currently working in the Long Range Penetrator (LRP) system program office. You are part of a team developing embedded software that will be used in the avionics system of the LRP. Your supervisor tasks you to use the SWCE module and a computer to perform several analysis tasks. These tasks are:

TASK ONE: Using the LRPBASE datafile, access the 'Adjust Parameters' function. Next, change the 'CPU Time Constraint' cost driver rating from 'High' to 'Very High'. Change the LRPBASE datafile code totals to reflect the following values:

NEW/MOD	(KDSI)	1987
REUSED	(KDSI)	1200
RETAINED	(KDSI)	585

After making these changes display the information on-screen. Proceed to task two.

TASK TWO: Save the information you entered in Task One in a new data file. Title this new data file 'LRPALT6'. Proceed to Task Three.

TASK THREE: You are now going to perform a sensitivity analysis on the 'CPU Time Constraint' cost driver you changed in Task One. Access the sensitivity analysis function and produce a graph of the 'CPU Time Constraint" driver with respect to development time. Adjust the graph to full screen size. Proceed to Task Four.

TASK FOUR: You are now going to perform a "What-If Analysis.' Access this function and select the LRPALT6 datafile for analysis. Also, select the LRPBASE datafile. Produce an activity distribution graph of the information in both of these datafiles. Adjust the graph to full screen size. You are now finished.

Appendix C: Competition Evaluation Model (CEM) Scenario

You are currently working in the Long Range Penetrator (LRP) system program office. You are part of a team acquiring the Inertial Navigation System (INS) for the LRP. You are tasked to investigate the effects of production competition on the INS acquisition. You decide to use a computer and the CEM module to help you in this analysis. You determine that you need to perform the following tasks:

TASK ONE: Create a new CEM data file and enter the following data:

NUMBER OF YEARS	5
FIRST FISCAL YEAR	88
DISCOUNT RATE	10

TASK TWO: Input the following data for the sole source producer:

FY 88 QUANTITY	50
FY 89 QUANTITY	200
FY 90 QUANTITY	250
FY 91 QUANTITY	250
FY 92 QUANTITY	250
FIRST UNIT COST	$100,000
PROGRESS CURVE RATE	92%
PRODUCTION RATE PARAMETER	100%

TASK THREE: Input the following data for the first source under competition:

Choose to split quantities by percentage

FY88 Production Quantity to the first source: 90%

FY89 Production Quantity to the first source: 83%

FY90 Production Quantity to the first source: 67%

FY91and FY92 Production Quantity to the first source: 60%

FIRST UNIT COST: $100,000

PROGRESS CURVE RATE: 92%

PRODUCTION RATE PARAMETER: 100%

TASK FOUR: Input the following data for the second source under competition:

Enter the same values for the second source that you entered for the first.

TASK FIVE: Enter the following data for the timing of shift and rotation due to competition:

NUMBER OF SHIFTS AND ROTATIONS: 1

CHOOSE TO ENTER SHIFTS AND ROTATIONS BY YEAR

FIRST YEAR OF SHIFT AND
ROTATION: 89

FIRST SOURCE-DOWNWARD SHIFT: 7%

SECOND SOURCE-DOWNWARD
SHIFT: 7%

FIRST SOURCE-STEEPER
ROTATION: 5%

SECOND SOURCE-STEEPER
ROTATION: 5%

TASK SIX: Input the following data concerning cost minimization, non-recurring costs, other recurring costs, and inflation rates:

Do not allocate larger competitive quantities to lower cost producer.

Choose to enter non-recurring costs.

Choose to enter non-recurring costs for all years.

Enter non-recurring costs of $5 million in FY88 and FY89. There are no non-recurring costs in later years.

Choose to not enter other recurring costs

Choose to not add inflation rates

TASK SEVEN: Perform the following operations:

Save the data file and give it an appropriate name

Display the data and check to see if they were entered correctly. Correct as necessary.

View and end the analysis.

TASK EIGHT: Make the following changes to the data file that you have just created and saved:

FIRST UNIT COST-SECOND COMPETITIVE SOURCE: $110,000

PROGRESS CURVE RATE-SOLE SOURCE: 93%

NON-RECURRING COSTS:
FY88: $10 million
FY89 and all following years: $7 million

QUANTITIES:
FY91 Total Sole Source
Quantity: 250
Quantity Given to First Source: 150
Quantity Given to Second Source: 100

MINIMIZE COSTS THROUGH QUANTITY ALLOCATION: Yes

TIMING OF ROTATION AND SHIFT: Change year to FY90

NUMBER OF ROTATIONS AND SHIFTS: Change to two with the first shift and rotation occurring at unit 50 and the second at unit 250.

ROTATION AND SHIFT: Enter 3 as the shift for both producers and 2 as the rotation.

TASK NINE: Save the changed data and name it appropriately. Quit the CEM module. You are finished.

Appendix D: Briefing Agenda

- Introduction of individuals involved
  - -- Researchers
  - -- Sponsor

- State the purpose of the briefing
  - -- Explanation of tasking (ASD's and AFIT thesis)
  - -- Introduction to the PMSS
  - -- Introduction to PMSS testing procedures

- Explain the reasons for doing this research
  - -- ASD reasons
    - -- Possible future use as a Decision Support System
      - -- Inputs from research can change the prototype
      - -- Inputs can influence the decision to use the PMSS
  - -- AFIT reasons
    - -- Fulfill thesis requirements

- Introduction of the PMSS
  - -- Developer
  - -- Purpose of the system
  - -- Structure of the system
  - -- Identify specific modules to be tested
    - -- Explain the purpose of each

- Explain testing procedures
  - -- Stress the system is being tested and not the user
  - -- Stress anonymity of the participant
  - -- Explain protocol collection concept
  - -- Explain use of a warm-up device

-- Explain the use of recording devices
-- Explain documentation and scenario preview
-- Explain debrief at the end of the testing session

- Solicit testing preferences and dates

- Answer any questions

Appendix E: Description, Preference and Ranking Sheet

The following PMSS modules are being tested. There is a brief description of each and a space for ranking the modules. Rank order each module with a '1' representing the module you would most like to help evaluate and '4' representing the module you would least like to help evaluate.

____ Contract Appraisal (CAPPS) Module - This module helps managers Monitor, Quantify, and Project contract status. It uses Contractor Performance Measurement (CPM) guidelines that are built into a database. CAPPS uses graphics almost exclusively unless the user specifies differently.

____ Software Cost Estimating (SWCE) Module - This module uses the Constructive Cost or COCOMO model for estimating software development costs. SWCE operates by determining the quantity of lines of code needed, applying 15 cost drivers to the problem, and giving estimated costs in man-hours and months of development time.

____ Government Activity Tasking (GAT) Module - This module assists with the planning, budgeting, and tracking of government activities. The program uses milestones to help keep track of financial and administrative aspects of government activity tasking on a stored database.

____ Competition Evaluation (CEM) Module - This module helps managers determine the economics of production competition. This is accomplished through the use of sensitivity analysis.

Fill in your two most preferred dates and times to help test the PMSS modules in the space provided below. The available days for testing will be

Mon/Wed/Fri, both morning and afternoon. You will be needed for about 1 hour, so please sign up at the start of an hour (i.e. 1300 and not 1330).

1st Date_____ 2nd Date_____

1st Time _____ 2nd Time_____

Appendix F: Test Session Checklist

1. Welcome participant. Check to see if they brought a copy of the scenario and program documentation. If they didn't provide them with a copy. Complete memorandum of agreement.

2. Introduce the participant to the computer game NYET by briefly demonstrating the game. Explain to the participant the reason behind asking them to play the game. Ask the participant to play the game.

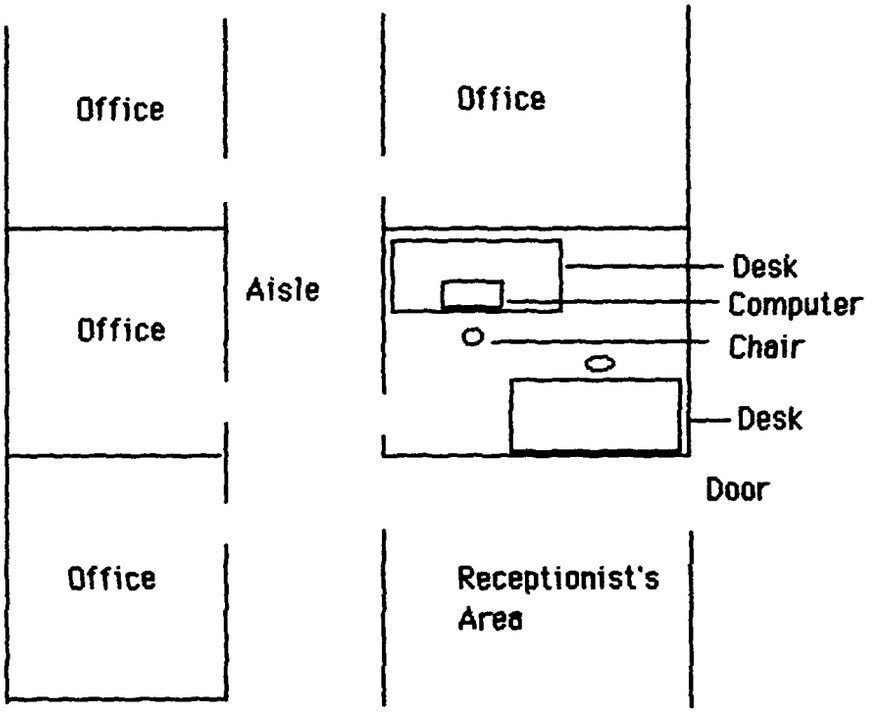
3. Ask the participant the following questions:
   a. What do you think of the on-screen explanations and arrangements of menu choices?
   b. What is your reaction to the choice of action keys?
   c. How do you react to the colors displayed on the screen?
   d. What do you think of the window arrangement on the screen?
   e. How do you react to the program messages?
   f. How does the screen resolution affect you?
   g. What is your opinion of the screen contrast?

4. When the participant feels ready, call up the PMSS module. Ask the participant to be sure to comment as they work the scenario problem. Explain that the researcher will try to minimize any comments or assistance during the test but will quietly monitor the test. Ask the participant to begin when ready and refer to the documentation as they need to.

5. When the participant has finished working the problem scenario ask them the following questions and record their answers on tape:

a. How much time did you spend studying the problem scenario and documentation before the test session?
b. Have you ever seen this PMSS module or any other PMSS module before today? If yes, explain your experience.
c. Have you ever taken any computer courses or training?
d. Do you frequently use computers on-the-job? Please explain.
e. Do you have any unique background or experience that might have affected your comments on the PMSS module? Please explain.
f. What feature did you like best?
g. What feature did you like the least?
h. If this PMSS module was available to you today, in finished form, would you use it on your job? Please explain your intended frequency of use and the primary purpose you would use it for.
i. What are the best and worst features of the program documentation? Do you think it can be improved? (If yes) How would you improve it?
j. Are there any other comments you would like to make?

Appendix G: Diagram of Testing Area



Appendix H: Test Session Transcripts

NOTE: S: represents the test subject R: represents the researcher

SWCE Respondent 1

TASK ONE

S: Alt Key, I'm not familiar with where that would be.
R: Right there.
S: Most of the programs I work with have direct control.
R: Would you comment on that?
S: Just the newness of it, I guess. Maybe thats just the 248 keyboard because I can't remember if there is an Alt key on the 100 system. I guess it just takes the place of the control key.
S: It seems like you should be able to come up over the name of the file and just hit enter, rather than go through this numbering.
S: I like to be able to use the keypad. It would be nice if that were available to you because you can do it real quick that way. It's just a personal preference.
S: I like that too when you don't always have to use 'enter', you can use the arrow to move down to your next thing. I like that type of a thing.

TASK TWO

S: It's nice when you save it, it just comes right up...no problem...it gives you the instructions to enter what file.
S: It would kind of be nice if there were some way that once you're in the file and you're doing some kind of analysis and you're through you'd have to put yourself out. It seems like this is automatically putting me out. I'm not sure if it did or not.

R: I don't think it did.
S: So, I'm still using that file?
R: Yes.
S: It would be nice if it had some key to get back to the main menu. Like right now I'm searching for a way to get back to the main menu. I'm not familiar with the module here.
R: Are you confused?

TASK THREE

S: Well, I'm looking at this fast track chart and I know I'm into the sensitivity analysis area so I'm looking under sensitivity analysis, trying to get to the 'what if' menu. Alt S...it didn't change for me. So now I'm saying, "Did I read it wrong, or push the wrong buttons?"

TASK FOUR

R: What did you push to get there?
S: Alt-R. It has both of them R and S. I'm not sure why it didn't respond to the S as well. That's the way I read the fast track chart anyway.
S: This is nice too. Like when you print a file out..you have the file name that you've done an analysis on. You have what type this thing is..like an activity distribution, but you still have an identifier for the file that you're working with. So, if you have several graphs together you're able to know which activity distribution you're analyzing there. I thin stuff like that is important to me to know which file you're working in. Maybe something in the corner that tells you you're still in that or maybe as you become more familiar with the module you know that you're in there.
R: But it would have been more reassuring to have a system message like that?
S: Right. Just like that. Just something real little in a corner or something like that. It seems like you have to keep coming back to the menu so something to let you know you're still in that file. All the information that

you're changing is in that file. That's very important. Keeps you from confusing information from some other file you thought you were in.
R: I'd like to ask you a couple of questions.
R: How much time did you spend studying the scenario and documentation?
S: I read it and tried to highlight it based on my experience with other programs what probably would be important. I spent approximately a half-hour just analyzing my task and what I'd be doing. I spent reading the manual...I read the instructions and they said if...whatever type of person you were...if you knew computers or had done software estimating stuff to read what sections. So I concentrated on those sections and probably spent an hour and a half reading.
R: Have you ever seen this module or any other PMSS module before today?
S: I haven't seen this one before or any of the others.
R: Have you ever taken any computer courses or computer training?
S: Yes. I've had BASIC programming training. I work with computers all the time where I work. We do software analysis with different cost models.
R: Do you have other software that is essentially doing the same thing for you?
S: Right.
R: Based on what you saw today, does this look any better than what you are currently using?
S: Looks very similar to some things that we're using. The graphics portion would be like an addition.
R: You don't have graphics now?
S: There are a couple of models that do but the ones that we use don't.
R: What software models do you use?
S: We have COCOMO, System 3, and PRICE-S.
R: Would there be any reason why you might prefer one over the other?
S: Actually, we like to get as many of them as we can because we can cross check things. It gives us more confidence in our estimating.
R: Do you have any unique background or experiences that might have affected your comments today?
S: All I would say is that my judgement is based on experience with other cost models.

R: Of the features you saw today, which one did you like the best?
S: I guess graphics...just because it's unique.
R: Which did you think was the worst?
S: Just right off the top of my head, the Alt key but that's just because it's different. I'm trying to think. I'm almost comparing this to WordStar. Its very powerful. When you start out you have this big menu and you have to keep going to other things. But once you learn it you can just zip right through it. When I first started with WordStar I thought it was very tedious but now that I've become fluent in its usage I just zip right through it. I see potential for that here.
R: At the novice level, do you think there are too many mnemonic commands?
S: I would say, yes, especially if this if just for program managers to do some things. They would be spending too much time trying to figure out what to do and where to go and it may defeat the main purpose of what they are trying to do.
R: If you were the programmer, how would you change the module?
S: I would put all the different variables that you can change and try to put them on one sheet and be able to adjust them however you want to adjust them. And after that be able to do different things like perform a 'what if' or if you want to do graphics on what you just did, and that kind of stuff.
R: In other words, a big spreadsheet?
S: I think the menus are okay. I guess easy access. Once you've change parameters easy access to any of the other type of things you want to do off of those changes.
R: If SWCE were available to you today in finished form, would you use it on the job?
S: Definitely. Because we like to use just as many as we can. The more we have the better our estimating can be. I definitely see a good usage for our office, today, if we had it. Everyone in our office would use it on a daily basis.
R: What did you think of the documentation?
S: I first thought they wasted a lot of paper on trivial things. As far as the

way its laid out with the index and table of contents, it was easy for me to find anything I needed. If I had a question, the way they described it and explained it seemed fairly understandable. I don't know how I would really improve it that much. Maybe if you had...like one of our software systems has a very simple, little pamphlet that shows the main highlights of the system. A quick reference card or something like that. You use something like that more than the manual. You refer to the manual when you have big questions or big problems but you're little reference is what you keep and use all the time.
R: Are there any other comments you would like to make?
S: Is it only useable on the Z-248s?
R: At the moment. Because of the graphics capabilities. Are you asking because you would want to run it on a Z-120?
S: We only have one Z-248 in our office and the others are Z-100 types. I don't know what the forecast is for getting other Z-248s but it would definitely limit us as to its usage. I can't think of any other comments.

SWCE Respondent Two

TASK ONE

S: When I was reading the user's manual I was trying to separate out what commands I could use. Some commands meant one thing in one part of the module and they mean something else in another part.
R: Do you have any specific examples?
S: In one place it said to graph you would use Alt-A and to adjust parameters you would use Alt-A also. Maybe I read it wrong.
S: I like that..'I'm working so don't worry'. That does help at least to know that it is doing something.

TASK TWO

S: I need to save now. Again, I like the delay screen. It shows that it is working.

TASK THREE

S: Okay. Now for sensitivity analysis.
R: Were you confused for a moment about how to get back from the full screen graph?

S: Yeah, I was confused since there aren't any...It didn't say on this screen or the other screen to press 'return' to return to the previous screen or the analysis portion. I think if I got used to it, it would be okay. But for someone who doesn't use it very much or a first time user it would be very easy to forget. For someone who doesn't know the normal workings of a computer would definetly have a hard time with it. I was a little confused at the beginning also as to how to graph it because when I moved from the first to the second, the graph seemed to change. Although it didn't. To choose what you want to graph as opposed to time or effort..that threw me at least for a while.

TASK FOUR

S: The only thing where I'm confused now is if I need to lead the data file again since I already have.
R: The data should still be there.
S: That's what I thought. Still I wasn't sure.
S: I didn't know you could graph two files together.
R: What do you think about that?
S: I like that. You can adjust them and compare them very easily. You can really get your point across with a graph, even if the person doesn't understand the reams and reams of data, the graph can really drive it home.

S: It would be nice, if up here in the corner, it said to press 'return' to go back to the analysis portion. I notice that this screen flashes. It flashes like its changing before it returns and it doesn't bother me but it did throw me at first because I wasn't sure what it was doing. I thought it had changed because I just saw that and didn't remember what I had looked at before.
R: Do you think that while it is changing the screen should disappear or what?
S: That would be good. Its like a cursor running all the way down through. It looks like it takes data and lays something else over it. Where it looks like it has changed the graph. It you don't notice that it hasn't you might think it has. Of course, with practice that wouldn't bother me at all. At first it did kind of throw me.
S: The menus are nice to have.
R: I noticed you didn't scroll through the help screens. Is that because you were already familiar with the documentation?
S: I was pretty sure I knew what to do from the instructions. I was pretty sure I could do the scenario with what I knew and what I had read. It was pretty well explained if you know what you are going to do.
R: Let me ask you some general questions. How much time did you spend studying the scenario and the documentation?
S: Basically, I kind of read through the scenario. I read through the manual in about an hour and that may be kind of a high estimate because I did have an interruption every so often. Then I read it through again this morning before I came..maybe a half-hour. The first time I went through it I tried to write down what kind of commands would help me. Kind of making a master list for myself so that in case I ran into trouble I could kind of look back. There I ran into the problem where in certain portions a certain command will do different things. Then I got further into the documentation and saw that the help screens were up there most of the time. So I decided I didn't need a master list that much.

R: What did you think of the documentation?
S: Well, I like how it was broken down into where user's who don't know a whole lot about software cost estimating should read through the whole manual.
R: Have you ever seen this or any other PMSS module before today?
S: I have known of the existence of CAPPS. I have never run it though. So the answer would be no.
R: Have you ever taken any computer courses or had any training in computers?
S: I took a couple of programming classes in school and then I used computers here pretty extensively. Not so much for analysis, like this module is for, but because I help with the computer resources in my organization.
R: How would you classify yourself as a user?
S: I would say moderate to advanced. I use them probably more than the average person but I'm not an expert.
R: In your job do you do any software cost estimating on the computers?
S: No I don't because I'm a program manager but this interested me because the program that I work on has a lot of software. So I thought that I'd like to see how this works.
R: Do you have any unique background or experiences that might have affected your comments today?
S: Not off hand. I can't think of anything unique. I have used Lotus 1,2,3 and even though this is not the same, I can see similarities.
R: Which feature did you like the best?
S: Obviously, I like the graphics. Most people would. It makes it a lot clearer. Instead of a bunch of numbers, you've got graphs.
R: So as a program manager, if your software people had data to present to you, would you rather have it presented graphically?
S: I would rather have it shown graphically with the numerical data to back it up, where I can go in more carefully to the parts I want to look at. The graphs are quick and easy to understand.
R: Which feature did you like the least?

S: I don't know if I could put one as the least. It's a little confusing having to load the data file but once I understood the overview where you load it and can change it, it falls into place. No glaring deficiencies that I can think of.
R: If this were available in finished form today, would you use it on the job?
S: I would use it on the job. I couldn't say that personally I would use it a lot but I would use it. Right now, where we are in our development, we could use it quite a bit. There are a few things I would change like how to return to previous screens or the sensitivity analysis menu or the main screen. A few little instructions would make it more friendlier. But all in all, it's pretty good. It's mostly polishing up thats needed.
R: Are there any other comments you would like to make?
S: Well, I can't think of anything right now. I noticed that I didn't use the instructions or the abbreviations. I could have used the instructions and taken more time to understand them. The comments I made might have changed because of that. I think its pretty easy to use.

SWCE Respondent 3

TASK ONE

R: Any feeling for the interface at the moment?
S: Well, it seems to be very smooth but the program might work better if it weren't within Symphony. I think, first of all, that that adds an extra cost plus there is a lot of overhead for that. I think they could probably come up with a very smooth one using say, DataEase or any of a number of methodologies or programs.
S: It seems to be quite sluggish, just to change a value. That should be very quick. I think that might be because of the recalculations done by the spreadsheet. They might be able to change that by telling it not to recalculate until it gets to a certain point in the program and then kick that in.

S: I like the use of colors. It makes it feel a little nicer.
R: Are those the colors that you like or would you prefer having different colors?
S: I think I would prefer having a higher contrast between what is changeable and what is not changeable. The light blue and the grey are quite close together.
S: Another thing that I wonder about is that they have this stupid red bar up there. Its missing something. There should be something written in it or not have it at all or make the whole window red.
R: I think that is intended to denote the cursor location.
S: OK
S: In general I don't care for the use of Alt keys. Too many keys. I think I would prefer special function keys where you could just hit one key.
S: In a way I kind of wonder about the instructions over here. Yeah, if you are a neophyte it helps you work through it the first time. But, if you are an expert you probably only want to see it if you call it up.
R: So you would like the help screens toggled?
S: Yeah. That would be better.

TASK TWO

No comments.

TASK THREE

S: Another thing too, is in this particular window, as they are describing it to you, the Alt-G key is used in sensitivity analysis, but then they go on to tell you what is going to happen on the next screen. Which I'm not going to remember, when I hit the keys for the next screen, what I'm supposed to do.
R: So you would like that portion of the instructions to come up on the next screen?
S: You're right. Because they say, "Select the Cost Driver," well, I don't have the choice of selecting a cost driver at this time. It would be the next window.

S: That was kind of weird. I hit that and the graph changed on me, which I wasn't expecting.
R: It didn't redraw the graph?
S: No, but the screen flashed. That's undesirable.
S: That's a little bit disconcerting to see it redraw the graph three time before it does what I want it to do. Its not nice.
R: What do you think of the display? Is it in the format that you would want that information?
S: Yeah, I think it has as much information and it shows it in as much detail. I like the fact that it doesn't start at zero where you lose resolution on the graph. I like that. And it does show you each point and where each point lies on the line. I like that. And it does do a real good job of labeling the axis.
R: What about the choice of colors here?
S: I prefer oranges and yellows myself. That's a personal preference. The line does stand out. The one thing you might want to have, if its possible, would be to bring these lines across which would give you...Like, I look at this point and I'm not sure, you know, just how close that is.
R: You would like lines across to both the horizontal and vertical axis?
S: Well, not the vertical in this case since there is such low resolution. But the horizontals would help you determine exactly what that point is.
S: It does not tell me how to get out of this, so I would assume a carriage return.
R: What do you think of that?
S: I don't like it. It doesn't tell you how to get out
S: Now once again it doesn't tell you how to get out of this. It has the Alt-D and the Alt-Z. I understand the constraints of the room on the screen. But, you need to have that kind of capability or at least something here like type in a question mark if you don't know what to do. Something to help you.

S: I'm not sure I like that. Now we have an inconsistent interface. Every time before I had a whole half a screen full of instructions, now I've got a full color menu. About a fourth of the screen. It should have something over here again if its going to be consistent. Or not have it at all ever unless I toggle it on. If I toggle it on it should have instructions there for you.
S: Okay, I want to do a "what if" and since I don't see anything that looks like "what if" I'll look at the instructions. They should help me.
R: That gives you the instructions for sensitivity analysis.
S: Right. Once again not telling me how to get back. I'm guessing though that Alt-M should back me up another menu but it does not. Return to main model, Alt-R, is what I should have hit.
R: Apparently you have confusion. How would you like to see that?
S: Well, I have seen it happen before where there is always one place on there, where its back-up to the next higher menu. And its consistently located in one place and its consistently the same command. And it gives you that idea. You know, "Return to Main Model," we've been talking menus and now its main models. Its somewhat confusing.
S: Okay, we want to do a what if analysis and it says to do an Alt-W. An activity distribution graph, thats Alt-A. A little bit disconcerting watching all that stuff flash.
R: Now what do you think of the choice of colors and the way the information is presented?
S: Well, I know that I would like to see file names always in caps. It makes everything look a little more uniform over here. It can be done by the user but it should be done by the program so there isn't any problem with that. I think that the light pastel type colors, the yellow and the light purple are close.
R: In the uses that you would make of this data, are colors relevant?
S: They are probably not so relevant as, rather you would want something with a little more contrast that will leap off the page better at you than

what this does. Now this is kind of ho-hum and just lays there, no excitement to the graph.
R: Now you've got color at the screen. You may or may not always have color at the printer.
S: I think, on the screen, if you've got color you should use the best choice of color. And then with the printer, most of us are not going to have color. Therefore, you should have patterns associated with those colors that would highlight the differences.
S: Adjust the graph to full screen. Alt-Z, now once again its only my experience that tells me that Alt-Z will do such a thing. This does not indicate that. The term "zoom", now that nice but if its going to make it full screen, and you've got room, say "Make full screen or full size." Something along those lines.
R: Zoom doesn't strike you as focusing in on some small part of the screen versus the whole screen?
S: Zoom in versus zoom out. Well, just reading it and knowing there is some function that makes it full screen I assume its not going to go in. Another thing too is that you're not really...A graph like this, it doesn't really make sense to zoom in. Your zoom will magnify. Zoom usually applies to zoom out. But, I really think, that more properly you should go to "Adjust to full screen." Something along those lines to make it a little better. It's redrawing the graph again four times which is disconcerting.
S: Another thing that might help, if they've got the capability, is it would make that a little sexier if they did a 3-D effect on it. A little shadowing or some depth to it. Keep it from being quite so plain. In this particular case I think the scale is a little too high, going every tenth. And it bothers me too that its months in thousands. You know, what is point one thousandths of a month? I don't like the scaling on it at all.
R: How about the way the axis are labeled?
S: Well, the bottom axis is fine. And even the labeling over here is fine but I don't like the scaling. You know, this should be one hundred months. It shouldn't be point one thousandths months. That has no concept. If you're going to scale it...scale it in years.

R: Does it bother you that you have to turn your head to read the label on the y-axis?
S: No, thats standard and it doesn't bother me at all. But you know, I think this is a little bit tight on the scaling and also I don't like the scaling in thousandths. I think there is room there to fill it out. You know they might have to be a little creative with how they scale it for a project that might make some sense. But I really think that if you're going to have months then you should scale it in years not months in thousandths. It should scale it in days, weeks, months, years, decades, you know, whatever. And once again there is nothing that tells me how to get back.
S: Alt-R. I like that. It once again gives you a consistency on what you call things. An overall impression of the system is that it seems a little bit sluggish. It does not seem to be consistent always as far as what its displaying and how it refers to things. And it has ho-hum displays and graphs. They could add a little excitement to it. There are places in there where they don't tell you what to do next and if you don't have some idea from past experience, you could be hurting. And considering that this is a Zenith 248, I feel that the module is sluggish, too sluggish for the average user. And I can imagine if you had a really big problem in there it would be much, much, slower. You might be able to get around that with a different spreadsheet program. Symphony is carrying a lot of baggage with it. You should be able to do it in Quatro or 1-2-3 , might be faster. It might be faster with a math coprocessor. But I wonder about the wisdom of putting it into a bigger system. I think you stand a pretty good chance of going into Turbo Pascal using some of the toolbox that they've got available. And you could come up equally nice displays and do it very quickly and very easily and it would be transportable. It would also work on much smaller machines plus the Government would have total rights to it. Symphony here..if you don't have the $410.00 for it, you can't use the module. I've never been that enamored of Symphony anyhow. I kind of feel that its...You know, if you need a spreadsheet, a database, or a word processor, why do I have to use theirs? Its a compromise on all of them. I'd rather use my Wordstar, my Quatro, my DataEase, to do the job.

R: Let me ask you some general questions here. Can you tell me approximately how much time you spent studying the scenario and documentation?
S: Oh, about 15 minutes.
R: Have you ever seen this module or any other PMSS module before today?
S: No. Well, wait a minute. There was a guy that came in and we sat around the table and he had a laptop and he flipped through some screens and stuff.
R: But nothing that would have affected your comments today?
S: Nothing that would have affected my learning curve.
R: Obviously, you have a background in computers and software. For the record could you explain what that is?
S: I have a master's degree in computer science from Wright State and all but a dissertation for a PhD in computer engineering out of AFIT. I have my own personal computer at home and I've been working with computers since college for my bachelors degree where I was the lab assistant and senior programmer for the school. And that was back in the 65-69 time frame.
R: What about the documentation? Any comments on that?
S: I didn't study it really thoroughly to be able to make a comment.
R: If this module was in finished form today, would you use it on the job?
S: I have no requirement for this on-the-job.
R: Do you have any other comments that you would like to make?
S: No. Just the ones I've already brought up.

SWCE Respondent 4

TASK ONE

S: I think the colors are kind of dull here. I tend to like lighter colors and more background.
R: Would you like the capability of changing colors?
S: Yes, I would.

S: It would be nice if the menu was in a different color than the instructions. It would help separate it.
R: Why did you enter the rating manually instead of selecting it from the menu?
S: I didn't notice the menu. Maybe if it were a different color it would stand out more.

TASK TWO

S: I guess it doesn't make any difference if I enter the name in all caps or lowercase?
R: No.
S: That save wasn't real quick. It takes a long time.

TASK THREE

S: (After drawing full screen graph) It needs to say 'Hit any key to continue'.

TASK FOUR

R: What do you think about the way the data is displayed on the graph?
S: Its very nice.
R: What about the way the axis are labeled?
S: It would be nice if the words were downwards so I wouldn't have to turn my head. It would also be nice if the titles were in a larger font.
S: It needs to be able to do cross hatching too. In case you don't have a color printer or plotter. We can't afford them. That would be something I would want. The main menu should be titled 'Main Menu' to differentiate between it and other menus. It confused me a bit earlier when the instructions I expected didn't come up.
R: Let me ask you some general questions. How much time did you spend studying the documentation and scenario?
S: About a half an hour to forty five minutes.

R: What did you think of the documentation?
S: I thought it was good. It would be nice if it had a tear-out quick reference card. Most people in my office can't find program manuals and if they find them they don't want to read them. They will try a new program and if they can't figure it out quick...that's it. They won't use it. It would be nice if you could get this to work with a mouse. I try to get away from the keyboard whenever I can.
R: Have you seen this or any other PMSS module before today?
S: No.
R: Can you explain your background in computers?
S: I majored in Systems Engineering and have a minor in computer science. I worked for a couple of years as a computer systems manager.
R: Would you classify yourself as a computer expert compared with the people you work with?
S: Yes. They consider me the resident expert.
R: Which feature did you like the best and which did you like the least?
S: I really liked the graphics if I can print it. If I can't print it, then the graphics are of little use to me. I thought it was easy to adjust the parameters. I thought that was very nice. I really didn't like having to use Alt-R prior to getting back to my main menu.
R: If this module were released in finished form today, would you use it on the job?
S: Yes I would.
R: Are there any other comments you would like to make?
S: No. I would just like to see it work with a mouse.

SWCE Respondent 5

TASK ONE

S: (Loading error occurs) This is not too bullet proof. Most of the programs that I've seen that use a spreadsheet as an overlay aren't very bullet proof.
S: They need to break out this paragraph. There is too much here. I can't

quickly scan. There is too much in this block. Short, as a matter of fact, don't use full sentences. It takes me too long to figure out what I want to do just from looking here. And most of the time, on these types of programs, you don't use it all day long. You use it, leave, come back to it, and use it again.
S: Okay, its loading it and it says 'Wait'. That's good.

TASK TWO

S: Alt-E for save. Why Alt-E, I'll never know. Evidently because there is no modularity to this. If I did an Alt-S it would do something else. Alt-S ought to be 'save' no matter where you're at.
S: Enter name. It is shown in all caps in the instruction sheet but I assume since this is DOS its not going to make a bit of difference.

TASK THREE

S: Again the instructions are very verbose.
S: Since the cursor only highlights part of the cost driver name its hard to tell whether this index is a two column list. That would be confusing to someone. Why doesn't it highlight both words?
S: (Draws graph). What does that show me? That shows me calendar manmonths on top which is really dumb. That graph should be flipped the other way. Everybody looks at the ordinate to have the months. And comparative values on the other axis.
S: Do you have to have Symphony to run this?
R: Yes.
S: That's a real bummer. That program is about as useful as a screen door in the summer.

S: I prefer to use function keys instead of key combinations. Too many keystrokes. I'd rather use one finger if at all possible. This would go really well with a mouse and it wouldn't be hard to write that driver. As a matter of fact, I think the Logictec mouse that is on the contract comes with a Symphony driver.
S: (Produces graph). Oh, isn't that pretty. Now what does that tell me? It has pretty colors. It would be nice if I could change the colors because yellow means something to people. In program management when you want to show them a color graph, green, yellow, and red are associated with risk. Now the other thing is hard copy. One of the things they've got to do with hard copy is when they go to print it out, you're not going to get colors. They've got to do the cross hatched version. So if I'm going to do this type of thing, I've got to be able to adjust the cross hatching as well. Part of the problem is, when you are pitching this to an audience, no audience is receptive to this in the same way. Therefore, part of the game in program management is knowing who you are pitching to and how to adjust your view graphs to suit them. An one of the things as part of the tools is not only does it do the analysis for me personally but it gives me the information I need to convince other people of my decisions. So, it is very important that I have the capability to adjust this. Or, if the capability can't be built into this (SWCE) output a file that can be read by Harvard Graphics, or Freelance, or Pagemaker, or something like that.
S: This presentation right here, if I was more experienced with it, looks like it might be okay. I have one gripe about this type of chart. I can't turn my head horizontal to read the vertical axis.
S: I can't remember how to get out of the zoom.
R: What do you think about that?
S: Well, after I sit here and talk to you a while I've forgotten how to do that. So, as a result of that it would be nice to have a DSU prompt. You always need a reminder because you are always worried about other things than how to use it. Although use of it will eventually get you to that. This is a good presentation for taking a quick look at things and playing 'what if'

and stuff. One of the best things they could do is to provide a DIFF file or something like it so you can make pretty graphs. They've got about 99% of it. My criticisms about changing graphs and stuff is in that last 1%.
S: I'm getting the information I need but it is constrained. It's constrained because its built as a series of macros in Symphony. I think it needs to be a program. You could code it in C. C is a good language and there are a lot of good programmers that know C. That would allow you portability from an MS-DOS system to a UNIX machine, to a VMS machine, to something else. Then what you would have is your virtual terminal drivers associated with that. Not everybody has a Z-248 on their desk.
R: Let me ask you a few quick questions. How much time did you spend studying the documentation and scenario?
S: Unfortunately, about 10 minutes.
R: What did you think of the documentation?
S: It appeared okay but I don't think thats a valid judgement because I didn't spend enough time with it. I should have read it cover to cover.
R: Have you ever seen this or any other PMSS module before today?
S: I have CASA.
R: Can you explain your computer background for the record?
S: I've been with the Government 18 years, 13 years of which was in Engineering and the remainder is program management. Engineering duties caused me to be involved with computers through simulations, steady state simulations of engines, fuel systems, and eventually for aircraft design, performance analysis, jobs, computer aided design, computer aided engineering. I was the system manager for one of the first VAXs that came out here so, to make a long story short, I'm intimately familiar with the scientific applications of computers. Second, as I was getting my M.B.A., I decided to take MIS courses to orient my tools to the management side. And so I feel fairly comfortable about knowing about MIS. I'm concerned with the user interface all the time because the damn things always frustrate people. And they always seem to pick the wrong things for our uses around here.

R: Which feature did you like the best and least?
S: I like the ability to change data and see the graphs. That tells you a lot. It seemed awkward to use because it is constrained to doing it with Symphony. I think it is a good proof of concept but I don't think they should have released it like that. It should be rewritten to make it easier to use. Error condition handling is not very good.
R: If this module were in finished form today, would you use it on the job?
S: I would but I would have great trepidation about recommending it to anyone who was not a computer user or at least at the moderate level.
R: Are there any other comments you would like to make?
S: I'm going to go back and investigate this further. This has got me interested. Maybe I can send some feedback to the people who are developing this. Because we don't have enough tools. The thing I'm doing is seeing if I can't get some of that good AI technology, that we are spending big bucks on, and put it into the way we do business.

SWCE Respondent 6

TASK ONE

R: What do you think of the 'I'm working' message?
S: It keeps the user busy. If you don't have anything on there then they get anxious. Especially, a person who doesn't use computers a lot.
S: I'm used to the system highlighting the name of the file you want to retrieve rather than just putting a number next to it.
S: I'd prefer a toggle to turn the instructions on and off. The trouble is they are good for the person who hasn't used the program much but they get in the way of the person who uses it once or twice a week or so.
R: You are about to change the driver rating by typing in 'Very High'. Did you notice the other menu options on the screen?
S: No. When you have two menus on the screen at the same time it gets very busy. A different color might help it stand out.

TASK TWO

R: What about response time?
S: About normal. I've seen a little better. I've seen a little worse.

TASK THREE

S: This module is based on Symphony. It should be a stand-alone program. I don't need to have to buy more software just to run this program. Also, the commands should only require one button to push instead of two.
S: I don't like the program redrawing the screen several times. Its confusing. It might even be better if it started out with a blank screen. I like the idea where you can seen it before you zoom it but the redrawing doesn't work real well. You start to think...Is that line changing? I didn't change anything. I'd rather the screen went blank and then the graph came up.
S: What would be nice would be a menu at the top of the zoomed screen or instructions to tell you how to get back.

TASK FOUR

S: I like the bar graphs. I would like to change the graph to different types: line, trend, area and that kind of thing. Some people prefer different types. It might be better, also, to be able to show them in cross hatching if you don't have a color printer. For color, I'd like to be able to change the colors. These don't offend me but different people like different colors. It also needs some text editing capabilities for tailoring the graphs.
R: Let me ask you some general questions. How much time did you spend studying the documentation and the scenario?
S: About 30-40 minutes.
R: Have you ever seen this or any other PMSS module before today?
S: I've seen CAPPS.
R: Could you explain your background with computers

S: I worked with computers in college doing batch processing. The only official software class I've had is with Lotus. I use computers daily on the job.
R: Do you have any unique background or experiences that might have affected your comments today?
S: I've been trained in several other software cost estimating packages and a lot of my comments came from what they've got. PRICE-S is one and System 3 is another.
R: Which feature did you like the best and which did you like the least?
S: I like the graphics ability best. There is nothing I didn't like except that all of the screens are real busy. What confused me the most was when I was in the input screen (Task One) and the output was on the same screen. There was nothing to highlight the output.
R: If this module were available in finished form would you use it on the job?
S: If I had other uses for Symphony other than just this, I would use it. I'm always looking for software so I can get two or three cross checks.
R: Are there any other comments you would like to make?
S: I can't think of anything else.

SWCE Respondent 7

TASK ONE

S: When I first read the documentation it seemed very complicated to me. They should have hired an English major to write it.
R: What did you think of the opening instruction screens?
S: I thought they were okay. I'm somebody who learns how to use the computer by having somebody tell me. Then when they tell me and I do it then I remember.
R: What did you think of the 'wait, I'm working' prompt?
S: I thought it was okay. I need something to tell me that because I'm very impatient. That is why I don't read documentation if there is somebody around who knows how to use it and I can ask them quick.

TASK TWO

R: The computer is saving the data now. What do you think of the response time?
S: I think it is very pokey.

TASK THREE

R: You selected the driver with respect to development time but now you appear confused?
S: Yes. I don't know what to do next.
R: You should tell it to draw a graph.
S: Oh, tell it to draw a graph.
R: What do you think of this?
S: I want it to tell me what to do next. I know that I can enter Alt-H for instructions but I want it to be friendlier to me and save me time.
R: When you scrolled through the drivers the graph window flashed. What did you think of that?
S: It freaked me for a minute.
S: How do I get out of here? (full screen graph). I tried Alt-M because I thought that would get me back. I want something on the screen to give me choices.
R: Just hit 'return'.

TASK FOUR

R: Does this graph look like something you would do on you job?
S: Yes, if you're doing a 'what if'. Bar graphs are helpful. We are now using PRICE-S and COCOMO software. I know this is COCOMO based so it would be very helpful in doing cross checks. This may be easier than our COCOMO. I don't know.

R: If this module were available in finished form would you use it on the job?
S: We will look at it very closely. We do need something else besides PRICE-S for cross checking.
R: How much time did you spend studying the documentation and scenario?
S: Just about an hour.
R: Have you seen this or any other PMSS module before?
S: I've never seen this module. I've seen CAPPS and CEM before.
R: Have you ever had any computer courses or training?
S: I had an ENABLE class and I had FORTRAN.
R: Do you use a computer daily on the job?
S: Yes, I do.
R: Do you have any unique background or experiences that might have affected the comments you gave here today?
S: I think the thing that affected my comments is my own ignorance. The only thing I can comment on is how it looks to me. I want something that is friendlier and I don't want it to waste my time. I don't want to have to go back to manuals and read them. As inexperienced as I am, there are many others more inexperienced.
R: Are there any other comments you would like to make?
S: The crowded text screens bothered me. I would like to see more white space. I find it real difficult to read things like that.
R: Which feature did you like the best?
S: I liked the graphics from the point of view of being helpful and letting management see some things like trends.
R: Which feature did you like the least?
S: Just that they don't have the screen menus to tell me what to do next. I'm used to Lotus and ENABLE and they constantly have something on the screen to help you. The documentation isn't consistent. They tell you how to do something in one place but when they mention it later they don't repeat how to do it. Just because they told you once they shouldn't presume that you remember. The way it is set up though is good. You don't get discouraged. The screen reproductions are good.

CEM Respondent 8

TASK ONE

No comments.

TASK TWO

R: Let me ask you, did this confuse you at all? The fact that you have to enter the figure in millions?
S: No, as long as it was pointed out. Somebody not knowing this, for the first time, would have to do a double take and make sure they read this and know that it had to be entered in millions. It may be easier to just type out the entire number. This way you wouldn't have to think twice, a lot of times, in case you miss this.

TASK THREE

S: Again, its nice not having to enter percents and stuff like that.
R: You seem to be confused a little bit.
S: Yeah, I was just going back to here and I don't remember that data or value. When I saw the next statement, 'First unit cost', I know where we were.
R: Would it have been better if that statement was followed by the existing value?
S: Yeah, that would help. It would remind you of what was happening.

TASK FOUR

R: Again, you took that route versus the carriage return.
S: Yeah, that true. I guess doing this for the first time, I'd rather go through all the steps and see that before I really got familiar with it so that I can rely on the data being in there already and just hit carriage return.

TASK FIVE

S: It may be a little confusing when it says, "Downward Shift (% or carriage return if none)'. It may be a little clearer if it said after percent " if none, carriage return".
R: Are you saying that this might lead you to strike the percentage key?
S: That or, I had to think twice exactly what that wanted. It might be simplistic but if it was phrased a little differently.

TASK SIX

No comments.

TASK SEVEN

R: Now what did you think of that menu for saving your file?
S: I'd rather have 'S' for Save. 'File out' is a little different. I've never seen that before. The reason I remembered 'F' is having gone through this once.
R: Having gone through the documentation?
S: Yes.
S: Any particular name?
R: Whatever you want.
R: What do you think of the way the data is displayed there?
S: I like the way it is displayed with the headings and by year giving totals. It might help to have dollar signs or something. One dollar sign or some kind of indication that this is cost. By just looking at the data there is no indication that it is cost here. It's laid out nice though.

TASK EIGHT

R: Let me ask you a question about your file name. You remember it because you just put it in but what if you did it in the morning and came back in the afternoon to make a change, would you remember the file name?
S: Probably, but I would probably give it a different name. Maybe less letters or applicable to what I was doing. I like giving file names that don't have a maximum number of letters or spaces.
R: What if you forgot the file name?
S: I have a habit of logging my data anyway elsewhere. But if there was a directory command I would use that if I was confused, to find out.
R: It brings up a question, you skipped one year there?
S: Yeah, accidently.
R: You can't go back to it. What do you think of that?
S: That I don't like. I would rather have the ability to go back to that. I knew I couldn't go back to it, that why I didn't try. I would like to have an 'Undo' command that brings me back to that position so I can do that.
S: Can I just go back now and change it?
R: You can if you want to.
R: Do you see what you did?
S: Right now I'm a little confused. I assumed that just by hitting return it wouldn't change anything, except what I inputted.
R: How many years are you changing?
S: One year. Year one. OK. I don't know how to get back to that other command.
S: Yeah, I'm stuck right now.
S: If there was a way to get me out of a current menu, quickly up to the very beginning
R: Of course, the old 'Control-Alt-Delete' will always work.
S: (Laughter). Yeah, let me get out of this and boot it up again.
S: It's confusing, what do they mean by year one.
R: One year.
S: Then I would just type in that year?
R: Yes.

S: I guess when I originally saw that, that it meant Year One, 88.
S: Yeah, when I first saw that I took it to mean to first year.
R: How would you like that displayed differently?
S: I guess that I would explain it or phrase it differently.
S: Shift rotation, choose a number between one and one. I guess I don't know what that means.
R: Thats because you only have one shift and rotation. If you had two or three shifts, it would have a range of values.
S: Okay.
S: Originally, I hit the wrong choice, rotation percentage and shift.
R: Which brings up a point..you have a fairly long menu. What do you think?
S: Yeah, I guess I would like to see it in a different format. It just kind of runs on.
R: Like a vertical format?
S: Yeah, maybe vertical would help or spaces in between or something that helps me identify it quicker and go to what I want.
S: That would be the biggest problem I would see. You're going to get into a menu and make some mistakes. You've got to be able to get out of it and correct it on the spot without having to repeat the whole program. Only reading through the documentation once, I don't understand all the commands.
R: That brings up a point about the interface. When you have stuff on the screen that you don't understand, would you like to see a help file?
S: Yeah, maybe a capability of stopping where you are now and going to a help file and finding what you're having trouble with and reading up on it and to go back quickly. That would help.
S: The biggest thing as far as being able to work it is to be able to correct something you've just done.

No comments.
R: Okay, well you've worked the scenario and I have some general questions I'd like to ask.
R: How much time did you spend studying the problem scenario and documentation?
S: I read it this morning for the first time and spent about a half-hour.
R: Have you ever seen this module or any other PMSS module before today?
S: No.
R: Have you ever taken any computer courses or computer training?
S: No.
R: Do you frequently use computers on your job?
S: Yes.
R: A Z-248 or mainframe?
S: Back in RW I used to use one. I don't have a 248 now. It was mostly for spreadsheets.and word processing.
R: Do you have any unique background or experiences that might have flavored the comments you made today?
S: Just my experience with what I'm used to. Different word processing programs. Going through that. And using the ASD AMS network and things like that.
R: Which feature did you like the best?
S: Its nice being able to input data like this and then being able to come up with results in a nice easy format as far as looking at it goes. I've never seen this program before. I'm a Contracting Officer and I wouldn't see this thing. Its pretty easy as far as inputting goes.
R: How about a worst feature?
S: The way the menus are laid out. I would like to see it more self-explanatory so I don't have to guess. Second, being able to undo what you just did. You've got to be able to do it.
R: What did you think of the documentation?

S: A class would be nice to be able to explain terms and what they mean. But, overall I think its pretty good. Some terms needto be elaborated on especially for a novice.
R: Are there any comments that you would like to make that you haven't previously thought of?
S: Not that I can think of. It appears to be a useful module as far as giving approximations of the usefulness of competition. Its fairly simple but again I would elaborate on the menu options and being able to recover when you've made a mistake. That the biggest two things.

CEM Respondent 9

TASK ONE

R: What do you think of the opening instruction screen?
S: Fine, just fine.

TASK TWO

R: Before you hit the enter key, let me point out that the first unit cost is in millions.
S: Oh, it didn't say that. So its point one.

TASK THREE

R: What do you think of the menu presentation you have there?
S: That's fine.
R: Do you like them in a horizontal format?
S: Yes. I think that horizontal is easier and for someone who is not a typist it is easier to follow in the horizontal.

TASK FOUR

S: 'CR'. What keyboard function is that?
R: That means carriage return.

TASK FIVE

No comments.

TASK SIX

No comments.

TASK SEVEN

S: We don't have a 'Save' on there.
R: We do. But its called 'File out'. How does that strike you?
S: I don't like it. I do just about everything with WordStar and it has 'Save'. That's what I'm used to seeing. I'd rather have 'Save and resume', or 'Save' period.
R: How does that display strike you?
S: I like the way it comes up here.
R: That's something you can use on the job?
S: Oh yeah, Oh yeah.

TASK EIGHT

R: You see what happened here is that these are hierarchical menus. You must end one to access another. What is your reaction to that?
S: That is confusing to me. Maybe when you get to every third choice you could end and go back to the menu. I don't know what that is called but it would make more sense.
S: The way that this menu is one long sentence makes it difficult.
R: How would you like to see it.

S: Maybe a vertical format. Not real deep. You know, maybe three here, three there, and three there. Like a lot of the other computer programs when they have the menus at the top.
S: Enter shift between 1 and 1? Boy, thats confusing.
R: That's because you only entered one shift originally.

TASK NINE

S: Top, top, that doesn't make any sense.
R: It's (S)top. Is that confusing?
S: To me it was. It would be better to read (S) Stop.
R: Let me ask you if you noticed that once you've entered a value and hit the return key, you can't scroll backup to the value to change it? What do you think of that?
S: Yeah, I noticed it and I don't like it
R: Let me ask you some general questions now.
R: How much time did you spend studying the documentation and the scenario?
S: I spent about 30 minutes before I got here and another 30 minutes yesterday. Probably a little over an hour.
R: Do you do this type of operation on your job?
S: No, not at this time.
R: Have you ever seen this or any other PMSS module before today?
S: No.
R: Have you ever taken any computer courses or computer training?
S: No. The only two I've taken are WordStar and Harvard Graphics.
R: Do you frequently use a computer on the job?
S: Yes, we use it quite a bit but its DataEase and that how we build a database.
R: This is on a mainframe?
S: Its on a mainframe.
R: Do you have any unique experience or background that might have affected your comments?

S: No.
R: Which feature did you like the best?
S: The major data displays.
R: Which feature did you like the least?
S: I think the menu choices being horizontal instead of vertical.
R: If this module were available in finished form today, would you use it?
S: We wouldn't, in this office, but I know of others who would.
R: What did you think of the documentation?
S: I guess the acronyms. They drive me crazy because I'm not a computer nut.
R: Are there any other comments you would like to make?
S: No. I can see where this could be a real useable tool in the program offices at ASD.

CEM Respondent 10

TASK ONE

R: What do you think about the opening screen?
S: Sort of long. Maybe if they had the titles in different colors it would liven it up.

TASK TWO

R: What do you think of the format for entering dollars?
S: I think it is confusing having to enter them as decimals.
R: In your business do you deal with acquiring items that are less than one million dollars or are they usually greater than that?

TASK THREE

S: More than a million. It's still confusing. I get most of the description in the manual but they don't put anything on the screen. It just jumps right into the model.
S: What do I do here. In the other one they had the percent sign there. This one doesn't have it.
R: You don't have to enter it as a decimal.

TASK FOUR

S: This is pretty nifty!
R: Do you like that function? (entering a carriage return to place a previously entered value in another portion of the program).
S: Yeah, unless of course you didn't have the same numbers. It's nice because you can't put a typing error in.

TASK FIVE

No comments.

TASK SIX

No comments.

TASK SEVEN

S: File out?
R: That's what you would think of as 'Save'.
S: Okay. That's strange. Like you're in a file and it should take you out.
R: What do you think of that?
S: Now if you had 'Save.'
R: You'd rather have 'Save'?
S: Yeah.

R: What do you think of the menus being in a horizontal format with the first letter of the word being used as the command?
S: I have no problem with that. It makes it easy. It's the word choice, like 'File out.' Another is 'Analysis'. I guess I'm more familiar with 'View".
S: Okay. I'm ready to display my data.
R: How do you like this display?
S: Since you have a color display, you might use the opportunity to use the color. Other than that it looks fine.
R: Does it look like something you would use on your job?
S: Yes.

TASK EIGHT

R: If you had a lot of data files, how would you remember the name of this data file?
S: You wouldn't. I can see a fault with that. I think...like a Lotus thing..where you have files to select. Where you can access the one you want without going to the directory.
R: What do you think of that menu?
S: It's a little busy. This one might be nice in a vertical format. One line is fine but having three is a little too much. You would kind of get lost.
S: I see something here. You have choices A,B, or C. I mean you get so used to pushing the first letter of the command.
S: Here, on this one they don't have the percentage. They aren't consistent.
S: On this one they don't have space between the questions. I don't know where my question is. I've lost my place.
S: This confuses me. Year (1). I read that as the first year. Maybe if they asked for the specific year.
S: This confuses me. Enter shift number between 1 and 1.
R: Earlier we told the program there was only one shift. If there were five shifts, it would say 'enter shift number between 1 and 5. What do you think of this when you only have one shift?

S: It's confusing to me. It would be nice to know what you had, spelled out, even if it were more wordy.
R: What do you think of using a menu prompt system to make data changes rather than a spreadsheet format where you could move to any cell?
S: It would be nice to have it in front of you. That's a lot of prompting to go through to make a change. Also, you could see it in the full perspective. The prompts help to lay the foundation when you set it up.

TASK NINE

S: Q for quit. They used E for end before. There is no consistency here.
R: Let me ask you some general questions. How much time did you spend studying the documentation and the scenario?
S: I think about a half an hour.
R: Have you ever seen this module or any of the other PMSS modules before today?
S: No.
R: Have you ever taken any computer courses or computer training?
S: No, just my computer classes in college and that was programming.
R: Do you usually use a computer on your job?
S: Most of the time but not every day.
R: Do you have any unique background or experiences that might have flavored the comments you made today?
S: I don't think so.
R: Which feature did you like the best and which did you like the least?
S: I guess, the part I liked the least was that there was no consistency in there. It kept changing the different menu prompts. The data display was nice and was easy to read.
R: If this module were finished, would you use it on the job?
S: Yeah, if my job dealt with it.
R: What do you think of the documentation?
S: It was easy to read and it had the examples, which were nice. I'd prefer having help files on the screen also.

R: Are there any other comments you would like to make?
S: No.

CEM Respondent 11

TASK ONE

R: Any reactions to the opening screens?
S: They're long and use basic construction. Basic construction isn't very exciting. I would leave out the methodology and just leave in the purpose.

TASK TWO

R: Are the acquisitions that you do on your job typically in the millions of dollars.
S: Mostly, yes.
S: I don't like the fact that you can't scroll back up to correct a mistake.

TASK THREE

S: This menu is not clear. I'd put in an 'or' instead of a comma...percentage or quantity. I'd prefer the choices in a vertical format.

TASK FOUR

R: You have an option to use a 'return' to automatically enter the values. Do you choose not to use it?
S: Oh, I thought it was the print. I didn't understand what that was.

TASK FIVE

No comments.

TASK SIX

No comments.

TASK SEVEN

R: You appear confused.
S: I don't know what 'File out' is.
R: It functions as a 'Save'.
S: 'File out' is the 'Save'? I don't like that. I would have had 'S' for save. I didn't understand what a 'file out' is.
R: What do you think of the data display?
S: I think it works.

TASK EIGHT

R: What do you think of this menu?
S: Kind of confusing. First of all they started out going A,B,C,D. Then they switched to different letters. They should have been consistent all the way through with it. I don't like a straight line menu. They could have done it in a graph type thing.
S: It doesn't tell you if its in millions or what. Point one for first unit cost. It could be point one anything for the existing value.
R: You had to enter changed data through a series of menu prompts. What if you could change the data on the display by moving the cursor to the data you want to change?
S: That would be great. That would be much better.

TASK NINE

R: What are your general overall reactions?
S: It got confusing. If you made a simple mistake it gets real confusing because you've got to go and search through all your stuff. I'd prefer to have some on screen help files.
R: How much time did you spend studying the documentation and the scenario?
S: About 20 minutes.
R: What did you think of the documentation?
S: It wasn't really all that clear. Having reproductions of the screens would have been helpful.
R: Have you seen this or any other PMSS module before today?
S: No.
R: Have you ever taken any computer courses or training?
S: In college. One course.
R: Do you use a computer daily on the job?
S: Not daily. I use them but not every day.
R: Do you have any unique background or experience that might have affected your comments?
S: No.
R: Which feature did you like the best and which did you like the least?
S: I like the data display because it brought everything together. I'd like to be able to get that display at any point in the program. The one I liked least was that you couldn't keep the data display on the screen when you needed to make changes but had to go through menus.
R: If this module were in finished form would you use it on the job?
S: I don't know what I would use it for. We are not in the competitive phase but are on the production line. Since we've been buying them since 1953 there is not a whole lot to be changed.
R: Are there any other comments you would like to make?
S: Something like this would be useful in the development phase.

CEM Respondent 12

TASK ONE

R: What did you think of the opening screens?
S: Is was too gathered...too condensed. I mean you're used to seeing a more rolling effect.
R: Do you mean too much information on the screen?
S: Yes, too much information.

TASK TWO

R: Is this 100 million or 100 thousand?
S: As you noticed the program is set for millions but the figure we want to enter is 100 thousand.

TASK THREE

R: What do you think of this menu?
S: Its fine.
R: You chose to enter a first unit cost by typing in the numbers. You could have entered the number by hitting the carriage return. Did you do it this way because you preferred it or didn't notice that option on the screen?
S: It was visible on the screen but I didn't notice it. It didn't stand out.

TASK FOUR

R: Now that you've entered the figures using the automatic feature, is this something you would use on the job or would you still prefer to enter the figures?
S: No, we would prefer to enter them with a carriage return.

TASK FIVE

No comments.

TASK SIX

R: Why did you enter a zero for non-recurring costs when you could have entered a carriage return for none?
S: It just didn't stand out for me to notice it.

TASK SEVEN

S: I'm looking for 'Save' and I don't see it.
R: 'File out' performs the same function.
S: I'm used to using 'Save'. What little familiarity I have with computers...'File out' means I want to get out of this file and whether I save it or not...I don't know.
R: What do you think of this data display?
S: Its not a bad display. I'm interested in seeing a breakout of the non-recurring costs and when they occurred. I'd like to see the information at the bottom of the display (first unit cost, progress curve rate, etc.) be located at the top. That's the way I'm used to seeing it.
R: What would you think of a capability of changing your data by bringing this display up and moving your cursor to the data you want changed rather than going through menu prompts?
S: I'd love that. I'm used to using spreadsheets and making changes by just moving the cursor up there and changing it.
S: You need more space down here at the total.

TASK EIGHT

R: You remembered your file name but would you have preferred to see a listing of your files to choose from?
S: Yeah, that would work.

R: What do you think of this menu?
S: My initial impression was that I expected to get my data display back.
S: Choose a rotation between 1 and 1. That leaves me blank.
R: That is because we have told the program there was only one rotation.
S: Its confuses me as to whether I've entered a shift or a rotation because they use the same instructions for both. Maybe if they said 'Enter rotation number', or 'Enter shift number', it would be better.

TASK NINE

S: They should have a 'Q' for quitting the module. You get used to using a 'Q' for quit and an 'S' for save when you use other programs. You know, its like a person gets used to putting on their shirt before their pants. Its hard to change.
R: What did you think of the large menu?
S: I wouldn't program it that way. You need some space between the choices and laid out vertically.
R: Let me ask you some general questions. How much time did you spend studying the documentation and the scenario?
S: Maybe two hours.
R: Have you ever seen this or any other PMSS module before today?
S: No.
R: Have you ever taken any computer courses or training?
S: None other than the college courses I've taken and self-training on the job.
R: Do you use a computer daily on the job?
S: No. Not daily. Sometimes.
R: Do you have any unique background or experience that might have affected your comments today?
S: No.
R: Which feature did you like the best and which did you like the least?
S: The graphics are pretty good.
R: The data displays?

S: Yes. The input procedures were pretty good. The only confusing part was with the rotations and shifts and the lack of consistency in the commands.
R: If this module were available in finished form, would you use it on the job?
S: I could see a use for it.
R: Are there any other comments you would like to make?
S: None that I can think of.

CEM Respondent 13

TASK ONE

No comments.

TASK TWO

No comments.

TASK THREE

No comments.

TASK FOUR

S: I can just carriage return right through these, right?
R: Yes. What do you think of that?
S: I think its great. It makes it a lot quicker.

TASK FIVE

S: I have to enter this by year, right?
R: Yes. Does that confuse you in any way?
S: No. I don't think so.

TASK SIX

No comments.

TASK SEVEN

R: What do you think of those menu choices?
S: I'd just prefer to see the word 'save'.
R: What do you think of the data display?
S: I guess I would prefer to see quantities going across the screen. Thats what I'm used to.

TASK EIGHT

R: You remembered your file name. What if you had forgotten it?
S: I would like to see a list of all the files come up. It might also be nice to have a time stamp with the file names.
R: What do you think of the error correction capabilities?
S: I'd like to be able to go back up to fix them.
R: What do you think of the menu layout?
S: I think it would be better if it were in color.
R: What do you think of the message 'choose shift or rotation between one and one?
S: It doesn't mean a whole lot. Its kind of confusing.

TASK NINE

R: Let me ask you some general questions. How much time did you spend studying the documentation and scenario?
S: I spent about two hours.
R: Have you ever seen this or any other PMSS module before today?
S: I think I've seen the CAPPS one.
R: Have you ever taken any computer courses or training?

S: I've taken a couple of programming courses.
R: Do you frequently use a computer on your job?
S: Yeah, I do. I probably use it a couple of times a week. Its a Z-150.
R: Do you have any unique background or experience that might have affected your comments today?
S: No.
R: Which feature did you like the best?
S: I think its fairly easy to read and understand what they want you to do. The simplicity of it.
R: Which feature did you like the least?
S: The one section on shifts and rotation. I know what a shift and rotation are but the program shouldn't have confused me. They need a help key so when you get stuck you can get help.
R: If this program were available in finished form, would you use it on your job?
S: I might use it to check the answers I would receive from my own model. I'd probably use it as a back-up.
R: The model that you are using now has been specifically designed for your application. Its not one of the main commercial models. Is that true?
S: Yes.
R: What do you think of the documentation?
S: The documentation is pretty good. It tells me basically what I need to know in order to operate it. I'm not really computer illiterate so I could probably get by with a lot less.
R: Are there any other comments you would like to make?
S: Not on this module, no. But I think its a good idea, what they are doing.

CEM Respondent 14

TASK ONE

R: I noticed you changed the screen to green. Did the white bother you?
S: Yes. The white is glaring.

TASK TWO

R: Any thoughts at this time?
S: Yeah, a lot of people won't pick up that that is dollars in millions.

TASK THREE

R: You appear confused?
S: Yes. It says enter appropriate letter. I don't see an appropriate letter.
R: Enter 'P' for percentage.
S: Obviously I didn't read it but I would prefer a clearer message.

TASK FOUR

R: You chose to reenter the same numerical values rather than just using the 'enter' key. Can you explain why?
S: It wasn't obvious to me. That is not a normal command or function so I just skipped to the obvious. What I know is going to work.

TASK FIVE

No comments.

TASK SIX

No comments.

TASK SEVEN

R: What do you think of using 'F' for file out?
S: Bad. I'd prefer 'Save'. Every other program you run you always look for a 'save' function.

TASK EIGHT

R: You remembered the name of your data file. If you had created the file in the morning and came back to it in the afternoon, would you still remember it?
S: Possibly not. I'd like to see an index on the screen.
S: This menu could really be lined up better.
R: What do you think of the error correction capabilities?
S: I obviously don't like it. I'd rather just move the cursor up and make the correction. Let's try it again.
R: It's taken two attempts to get your data in. What would make it easier for you?
S: A much clearer display of the menu. I ran across 'min cost' and went to 'other'. I expected to see a sub-menu there that would include 'min cost'.
S: 'Enter choice between one and one'. It should go straight to the year. There is no need to ask the question obviously.

TASK NINE

R: What do you think?
S: I don't like it. The menus aren't very clear and aren't laid out well at all.
R: Do you use something like this on your job now?
S: No. I just use Lotus.
R: If this program were designed in a spreadsheet form would you prefer it?
S: Personally, yeah.
R: How much time did you spend studying the documentation and the scenario?
S: Two minutes tops.
R: Have you ever seen this or any other PMSS module before today?
S: No.
R: Have you ever had any computer courses or training before?
S: No.
R: Do you frequently use a computer on your job?

S: Not up here. We will be soon. I worked for the Navy for a couple of years and we had Z-248s on our desks.
R: Do you have any unique background or experience that might have affected your comments today?
S: No. Nothing unique in any way.
R: What do you think was the best feature?
S: Thats a tough question. The content itself is so minimal. Grant you the program says it can't be used for decision making and its right. It seems very simplistic. I guess that's its advantage now that I think of it. You can make an initial determination that we should really dig into a second source and really get down into the guts of what it would cost.
R: What do you think was the worst feature?
S: Once again, obviously the menus.
R: If this module were in finished form would you use it on your job?
S: I use it as one of a library of things.
R: What do you think of the documentation?
S: I didn't spend much time on it. I have to give you a 'no comment.'
R: Are there any other comments you would like to make?
S: The purpose of the module is to use the output as one of many tools. I think you've got a good start on it. It needs cleaning up for ease of use. If the intent is to make it a primary tool, I think a lot of work needs to be done. It needs further detailed breakdowns such as non-recurring costs for tooling. I'd want further categories. Everybody likes to treat government as a sunk cost but if you split a program into dual sources, you obviously are duplicating a lot of efforts. Thats not inconsistent that this module doesn't take this into account because nobody does. Those are just a couple of examples.

Appendix I: Memorandum of Agreement

I want you to feel free to make any constructive comments during your PMSS test session. This freedom of expression will help me to collect the best possible research data. I assure you that any comments you give me will not be attributed to you in any way.

JOHN A. KANE, Capt, USAF
Researcher

I understand that I am free to make any constructive comments concerning the PMSS during my test session without attribution to me.

Research Participant

Appendix J: SWCE Sample Instruction Screen

AW1: A ' SOFTWARE COST ESTIMATING MODULE SHEET

MENU	
HELP & INSTRUCTIONS	
MENU	ALT-M
INSTRUCTIONS	ALT-H
ABBREVIATIONS	ALT-B
SWCE PARAMETERS	
ADJUST PARAMETERS	ALT-A
VIEW PARAMETERS	ALT-V
GRAPHICS ANALYSIS	
SENSITIVITY	ALT-S
WHAT-IF	ALT-W
SAVING & LOADING	
SAVE CURRENT DATA	ALT-E
LOAD DATA FILE	ALT-L
QUIT	ALT-Q

INDEX

SOFTWARE COST ESTIMATING MODULE
OPERATING INSTRUCTIONS

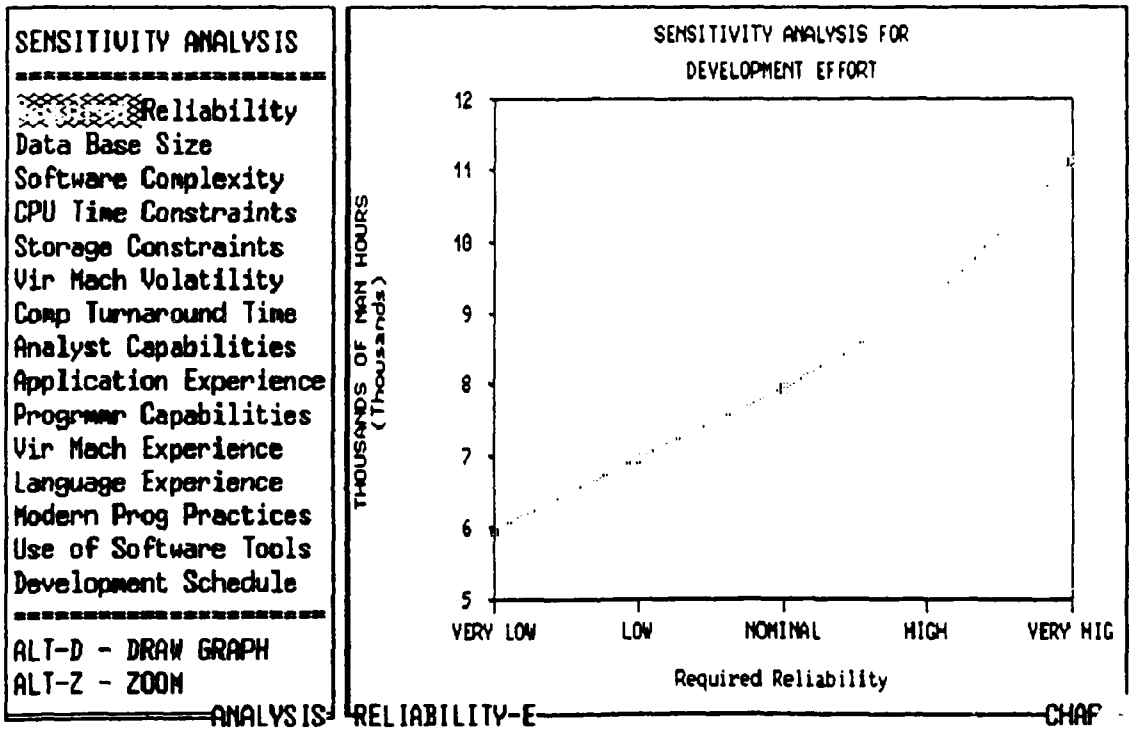
GENERAL
You are currently in the INSTRUCTION window. You can page through these instructions by pressing the <PgDn> or <PgUp> keys.

MENU
The Menu, shown in the left window, provides you with the options available for this module. To select a menu item, press the <ALT> key and the letter key (e.g., <A>) simultaneously. The menu option can be selected from any point in the module.

ALT-M MENU
Pressing the <ALT> key and <M> key will

INSTRUCTIONS

Appendix K: SWCE Sensitivity Analysis Driver Screen



Appendix L: CEM Data Summary Screen

Changed Data and Analysis

DATA SUMMARY

FY	***** SOLE	QUANTITIES 1ST C	***** 2ND C	O.RECUR COST	N.RECUR COST	INFLATE %
88	50	45	5	0.0	10.0	0.0
89	200	166	34	0.0	7.0	0.0
90	250	168	82	0.0	0.0	0.0
91	250	150	100	0.0	0.0	0.0
92	250	150	100	0.0	0.0	0.0
TOTAL	1,000	679	321	0.0	17.0	

FIRST UNIT COST	1.000	1.000	1.500
PROGRESS CURVE	93.00	92.00	92.00
PRODUCTION RATE	100.00	100.00	100.00
SHIFT %		7.00	7.00
ROTATION %		5.00	5.00
SHIFT %		3.00	3.00
ROTATION %		2.00	2.00

DISCOUNT RATE (%)= 10
ASSIGN COMPETITIVE SPLIT
TO MINIMIZE COST? YES
UNIT OF SHIFT/ROT= 50

UNIT OF SHIFT/ROT= 250

COMPETITION ANALYSIS

	SOLE SOURCE		********* COMPETITIVE			**********			DISCOUNTED -INFL%
		RECUR	QUANTITIES *		RECUR	O.REC	N.REC	COMPET	COMPET
FY	QTY	COST	1ST	2ND	COST	COST	COST	SAVING	SAVING
88	50	36.8	45	5	38.7	0.0	10.0	-11.9	-11.9
89	200	119.5	166	34	111.0	0.0	7.0	1.5	1.4
90	250	134.7	168	82	123.0	0.0	0.0	11.7	9.7
91	250	127.5	150	100	116.5	0.0	0.0	11.0	8.2
92	250	123.0	150	100	110.1	0.0	0.0	13.0	8.9
TL	1,000	541.6	679	321	499.4	0.0	17.0	25.2	16.2

Bibliography

1. Defense Systems Management College. The Program Manager's Support System (PMSS), An Executive Overview And System Description. Fort Belvoir, Virginia, January 1987.

2. Ericsson, K. Anders and Herbert A. Simon. Protocol Analysis, Verbal Reports As Data. Cambridge, Massachusetts: The MIT Press, 1984.

3. Shneiderman, Ben. Designing The User Interface, Strategies For Effective Human-Computer Interaction. Reading, Massachusetts: Addison-Wesley Publishing Company, 1987.

4. Rissland, Edwina L. "Ingredients Of Intelligent User Interfaces," International Journal of Man-Machine Studies,21: 377-388 (1984).

5. Carey, Tom "User Differences In Interface Design," Computer, 14-19 (November 1982).

6. Mozeico, Howard "A Human/Computer Interface To Accomodate User Learning Stages," Communications of the ACM,25: 100-104 (February 1982).

7. Davis, Roger L. Class handout distributed in IMGT651, Systems Analysis and Design. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, May 1988.

8. Hollands, J.G. and Philip M. Merikle. "Menu Organization And User Expertise In Information Search Tasks," Human Factors,29(5): 577-586 (1987).

9. Shinar, David and Helman I. Stern. "Alternative Option Selection Methods In Menu-Driven Computer Programs," Human Factors,29(4): 453-459 (1987).

10. Nickerson, Raymond S. "Why Interactive Computer Systems Are Sometimes Not Used By People Who Might Benefit From Them," International Journal of Man-Machine Studies,15: 469-483 (1981).

11. Williges, R. C. and B. H. Williges. IFAC Analysis, Design And Evaluation Of Man-Machine Systems. "Human-Computer Dialogue Design Considerations," Baden-Baden, Federal Republic of Germany, 1982.

12. Rushinek, Avi and Sara F. Rushinek. "What Makes Users Happy?," Communications of the ACM, 29(7): 594-598 (July 1986).

13. Ting-peng Liang. "User Interface Design For Decision Support Systems: A Self-Adaptive Approach," Information & Management, 12: 181-193 (1987).

14. Norman, Donald A. "Design Rules Based On Analyses Of Human Error,: Communications of the ACM,26(4): 254-258 (April 1983).

15. Lantz, Keith A. "On User Interface Reference Models," SIGCHI Bulletin,18(2): 36-42 (Undated).

16. Simpson, Henry. "A Human-Factors Style Guide For Program Design," BYTE, 108-132 (April 1982).

17. Apple Computer Inc. Human Interface Guidelines: The Apple Desktop Interface. New York: Addison-Wesley Publishing Company, Inc., 1987.

18. Durrett, John and Judi Trezona. "How To Use Color Displays Effectively," BYTE 50-53 (April 1982).

19. Tullis, Thomas S. "An Evaluation Of Alphanumeric, Graphic, And Color Information Displays," Human Factors, 23(5): 541-550 (1981).

20. Davis, Elaine G. and Robert W. Swezey. "Human Factors Guidelines In Computer Graphics: A Case Study," International Journal of Man-Machine Studies, 18: 113-133 (1983).

21. Ives, Blake. "Graphical User Interfaces For Business Information Systems," MIS Quarterly/Special Issues, 15-47 (1982).

22. Foehr, Theresa and Thomas B. Cross. The Soft Side Of Software, A Management Approach To Computer Documentation, New York: John Wiley & Sons, 1986.

23. Houghton, Raymond C. Jr. "Online Help Systems: A Conspectus," Communications of the ACM,27(2): 126-132 (February 1984).

24. Cohill, Andrew M. and Robert C. Williges. "Retrieval Of HELP Information For Novice Users Of Interactive Computer Systems," Human Factors, 27(3): 335-343 (1985).

VITA

Captain John A. Kane [redacted]

He graduated from high school in North Muskegon, Michigan, in 1969 and enlisted in the USAF in March 1970. He pursued a college education during off-duty hours and received the degree of Bachelor of Science in Occupational Education, in May 1979, from Southern Illinois University at Carbondale, Illinois. He received his commission in February 1980 after completing Officer's Training School. He was assigned to the Air Force Flight Test Center at Edwards AFB, California where he served as a squadron executive officer, Chief, Systems Management Branch in the Directorate of Administration, and as Director of Administration until September 1984. He then served as course chief and master instructor in the Administration Officer Management Course (Base DA), Keesler Technical Training Center, Keesler AFB, Mississippi until entering the School of Systems and Logistics, Air Force Institute of Technology, in May 1987.

[redacted]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GIR/LSY/88D-7	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Systems and Logistics 6b. OFFICE SYMBOL (If applicable) AFIT/LSY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) See Box 19	
12. PERSONAL AUTHOR(S) John A. Kane, B.S., Capt, USAF	
13a. TYPE OF REPORT MS Thesis 13b. TIME COVERED FROM _____ TO _____ 14. DATE OF REPORT (Year, Month, Day) 1988 December 15. PAGE COUNT 149	
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Interface, Man Computer Interface, Logistics Management, Management, Management Information Systems, Management Planning and Control
05	01		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: ANALYSIS OF THE HUMAN-SYSTEM INTERFACE USED IN THE PROGRAM MANAGERS SUPPORT SYSTEM: SOFTWARE COST ESTIMATING AND COMPETITION EVALUATION MODULES

Approved for public release IAW AFR 190-1.

WILLIAM A. MAUER
Associate Dean 27 JAN 1989
School of Systems and Logistics
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433-6583

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Dorothy J. McBride, Lt Col, USAF	22b. TELEPHONE (Include Area Code) (513) 255-4845 22c. OFFICE SYMBOL AFIT/LSY

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

This research analyzed the human-system interface used in the Program Managers Support System (PMSS) software. The PMSS software is planned to consist of 18 modules that will run as an integrated system on standard Air Force microcomputers. Six modules are presently in prototype form. This research analyzed the human-system interface of two of those modules, the Software Cost Estimating and Competition Evaluation program.

Problem scenarios involving software cost estimation and competition evaluation were developed with the assistance of Aeronautical Systems Division (ASD) experts. Fourteen ASD program management persons were chosen as test participants (seven for each module) and attempted to work the problem scenario with the prototype software. The participants were only permitted to view the scenario and program documentation prior to the test. Their critical comments were recorded during the test session and later transcribed.

Transcribed comments were categorized and compared to expert guidelines published in the literature. Finally, suggestions for improvement in the human-system interface used in these modules were drawn from this comparision